带 CAN 控制器的单片 8 位微控制器－P8xC591

1. 特性

1.1 P8xC591 的 80C51 相关特性

- 全静态 80C51 中央处理单元，可提供 OTP，ROM 和无 ROM 型
- 16K 字节内部程序存储器，可外部扩展到 64K 字节
- 512 字节片内数据 RAM，可外部扩展到 64K 字节
- 3 个 16 位定时/计数器 T0，T1（标准 80C51）和附加的 T2（捕获&比较）
- 带 6 路模拟输入的 10 位 ADC，可选择快速 8 位 ADC
- 2 个 8 位分辨率的脉宽调制输出（PWM）
- 作为标准 80C51 引脚时有 32 个 I/O 口
- 带字节方式主和从功能的 I^2C 总线串行 I/O 口
- 片内看门狗定时器 T3
- 扩展的温度范围：-40~+85℃
- 加速指令周期 500ns@12MHz
- 操作电压范围：5V5%
- 保密位：
  －ROM（2 位）
  －OTP（3 位）
- 32 字节加密阵列
- 4 个中断优先级，15 个中断源
- 全双工增强型 UART，带有可编程波特率发生器
- 电源控制模式
  —时钟可停止和恢复
  —空闲模式
  —掉电模式
- 空闲模式中 ADC 有效
- 双 DPTR
- 可禁止 ALE 实现低 EMI
- 可编程 I/O 口（准双向、推挽、高阻和开漏）
- 掉电模式可通过外部中断唤醒
- 软件复位（AUXR1.5）
- 复位脚低有效
- 上电检测复位
- Once 模式

1.2 P8xC591 与 CAN 相关的特性

- CAN2.0B 控制器，支持 11 位标准和 29 位扩展标识符
- 8MHz 时钟可实现 1Mbit/sCAN 总线速率
- 64 字节接收 FIFO
- 13 字节发送缓冲区
- 增强型 PeliCAN 内核（取自 SJA1000 独立 CAN2.0B 控制器）

1.2.1 PeliCAN 特性

- 4 个独立可配置的筛选器（验收滤波器）
- 每个筛选器有 32 位区分符

  －32 位 Match

  －32 位 Mask
- 每筛选器的 32 位 Mask 允许唯一的组寻址
- 更高层的协议支持标准 CAN 格式：

  －最多 4 个 11 位 ID 筛选器可筛选两个数据字节

  －即数据流可通过 CAN ID 和数据字节内容进行筛选
- 最多 8 个 11 位 ID 筛选器其中半数可筛选第一个数据字节
- 所有筛选器都可 " change on the fly "
- 只听模式，自检测模式
- 错误代码捕获，仲裁丢失捕获，可读的错误计数器

2 概述

PP8xC591 是一个单片 8 位高性能微控制器，具有片内 CAN 控制器，从 80C51 微控制器家族派生而来。它采用了强大的 80C51 指令集并成功的包括了 Philips 半导体 SJA1000 CAN 控制器的 PeliCAN 功能。

全静态内核提供了扩展的节电方式。振荡器可停止和恢复而不会丢失数据。改进的 1：1 内部时钟预分频器在 12MHz 外部时钟速率时实现 500ns 指令周期。

图 1 所示为 PP8xC591 的方框图。微控制器以先进的 CMOS 工艺制造，并设计用于汽车和通用的工业应用。除了 80C51 的标准特性之外，器件还为这些应用提供许多专用的硬件功能。

PP8xC591 组合了 P87C554（微控制器）和 SJA1000(独立的 CAN 控制器)的功能，并具有下面的增强特性：

- 增强的 CAN 接收中断
- 扩展的验收滤波器
- 验收滤波器可 " change on the fly "

PP8xC591 和 P87C554 之间的主要区别在于：

- 片内 CAN 控制器
- 6 输入 ADC
- 低电平复位
- 44 个引脚

3 订购信息

类型编号	封装	温度范围(℃)
P83C591VFA	PLCC44	-40 ~ +85
P87C591VFA	PLCC44	-40 ~ +85
P83C591VFB	QFP44	-40 ~ +85
P87C591VFB	QFP44	-40 ~ +85

4 方框图

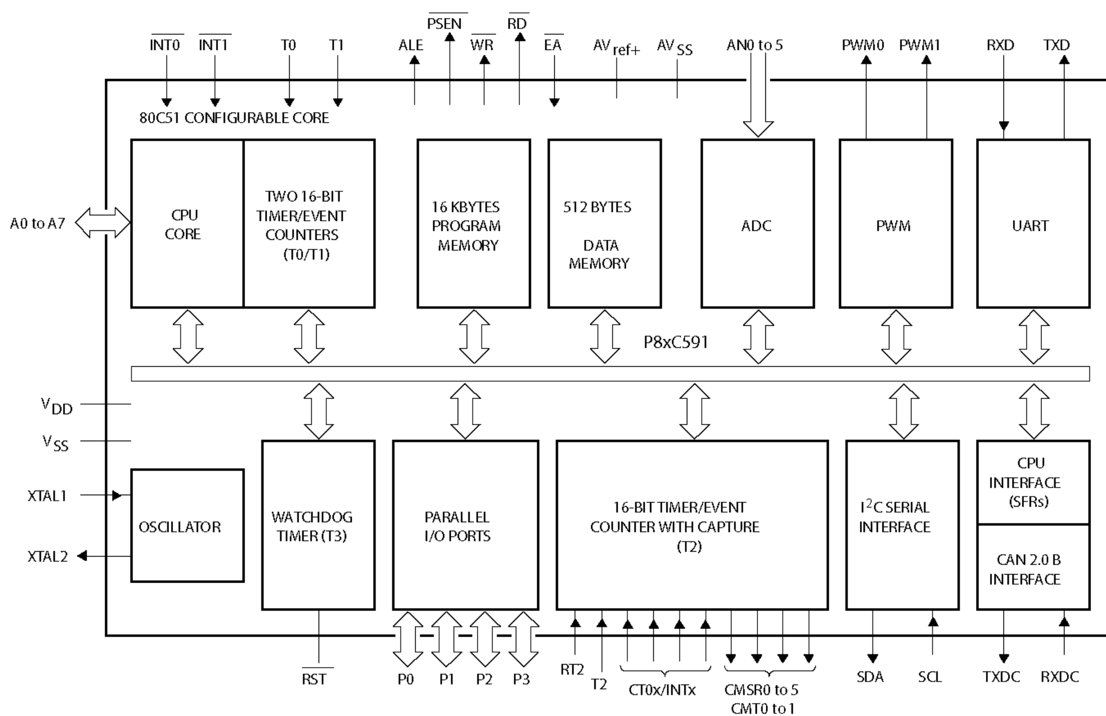


图 1 PP8xC591 方框图

5 功能图

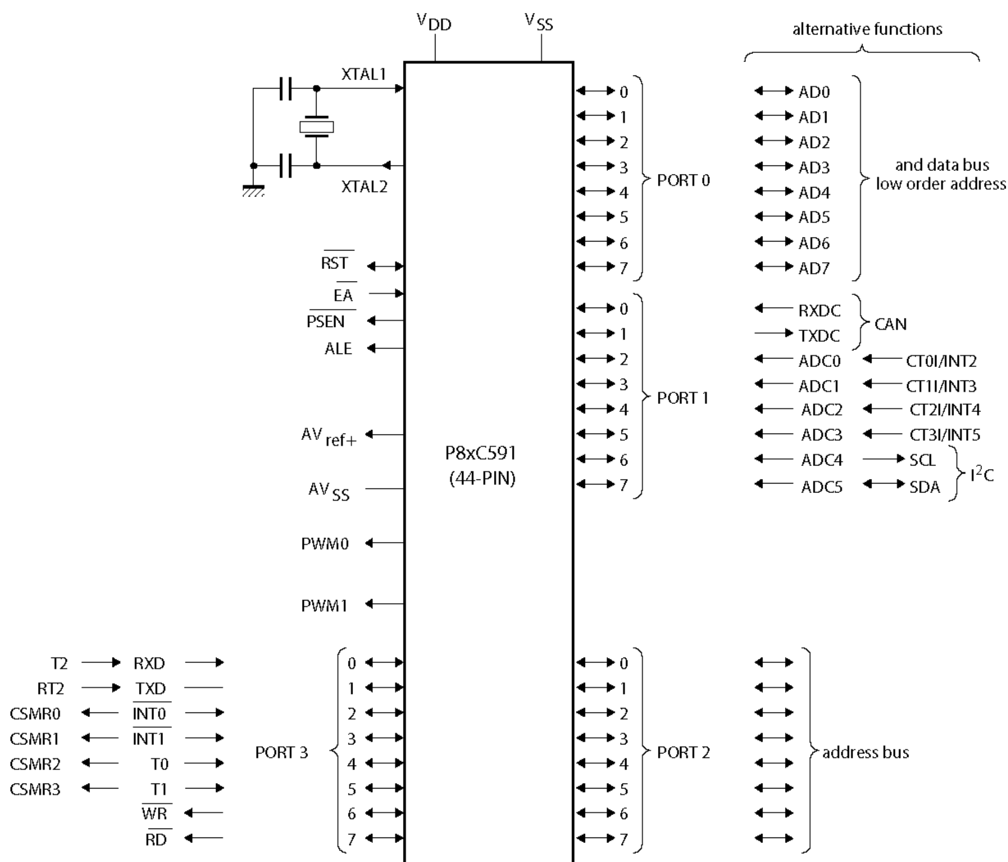


图 2 功能图

6 管脚信息

6.1 管脚分布图

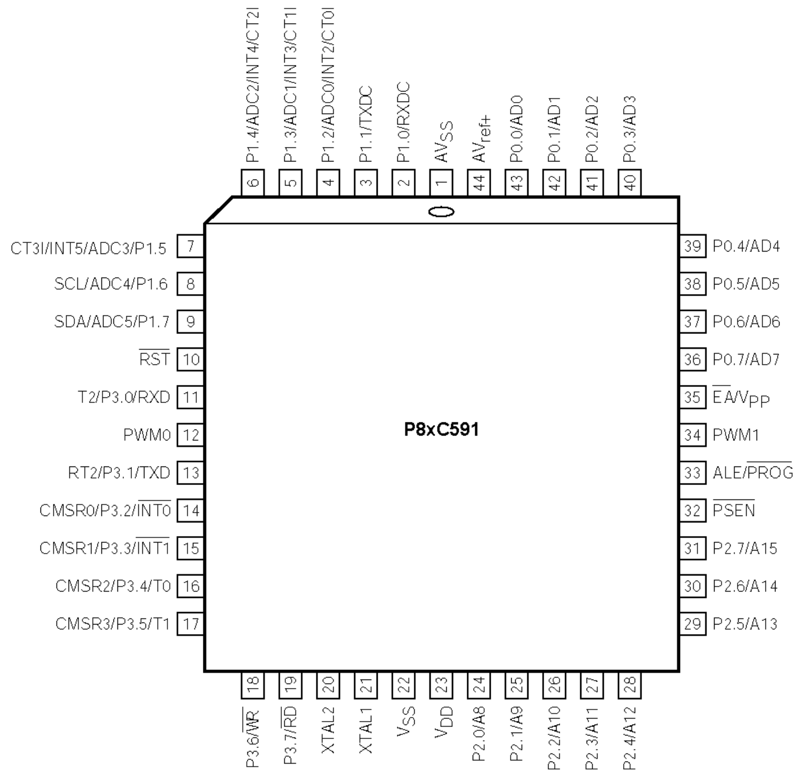


图 3 44 脚 LCC 封装管脚图

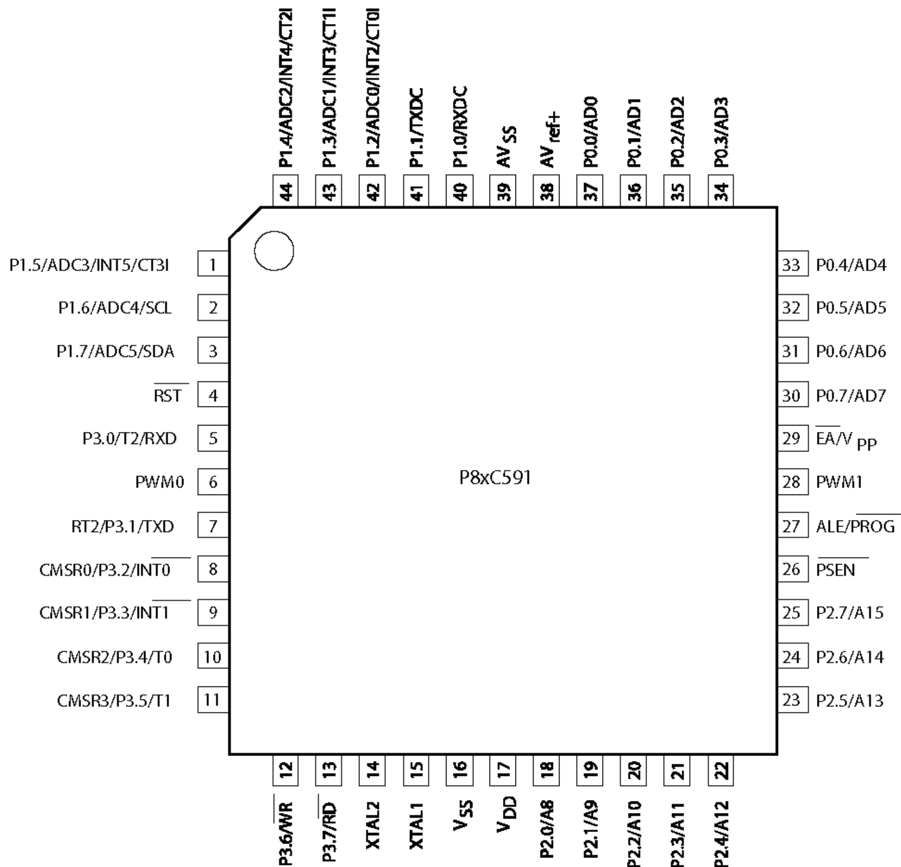


图 4 44 脚 QFP 封装管脚图

6.2 管脚描述

表 1 QFP44/PLCC44 管脚描述

符号	管脚号		描述
	QFP44	PLCC44	
RST	4	10	复位：PP8xC591 复位输入。当定时器 T3 溢出时提供复位脉冲输出
P3.0-3.7			P3 口：8 位可编程 I/O；P3 可驱动 4 个 LSTTL 输入。 P3 口还提供以下功能：
P3.0/RXD	5	11	RxD: 串行输入口 T2： 事件输出
P3.1/TXD	7	13	TxD: 串行输出口 RT2: T2 定时器复位信号。上升沿触发
P3.2/INT0/CMSR0	8	14	$\overline{INT0}$: 外部中断 0 CMSR0:定时器 T2 比较和设置/复位输出
P3.3/INT1/CMSR1	9	15	$\overline{INT1}$: 外部中断 1 CMSR1:定时器 T2 比较和设置/复位输出
P3.4/T0/CMSR2	10	16	T0: 定时器 0 外部输入 CMSR2:定时器 T2 比较和设置/复位输出
P3.5/T1/CMSR3	11	17	T1: 定时器 1 外部输入 CMSR3:定时器 T2 比较和设置/复位输出
P3.6/WR	12	18	$\overline{WR}$: 外部数据存储器写选通
P3.7/RD	13	19	$\overline{RD}$: 外部数据存储器读选通 复位时，P3 异步驱动为高 通过 P3M1 和 P3M2 寄存器可将 P3 口设置为 4 种模式之一： P3M1.x P3M2.x 模式描述 0 0 准双向（默认的标准 C51 配置） 0 1 推挽 1 0 高阻 1 1 开漏
XTAL2	14	20	晶振脚 2：反相振荡放大器输出，当使用外部振荡器时钟时开路
XTAL1	15	21	晶振脚 1：反相振荡放大器输入和内部时钟发生电路输入。使用外部振荡器时钟时作为外部时钟信号的输入端。
Vss	22	16	地：0v 参考点
Vcc	44	38	电源:提供正常、空闲和掉电工作电压
P2.0/A08-P2.7/A15	18-25	24-31	P2 口：8 位可编程 I/O 口 A08-A15：外部存储器高地址。还具有以下功能： 外部存储器（A08-A15）高地址字节。还可作为 EPROM 编程和校验时的高地址。 复位时，P2 异步驱动为高 通过 P2M1 和 P2M2 寄存器可将 P2 口设置为 4 种模式之一： P2M1.x P2M2.x 模式描述 0 0 准双向（默认的标准 C51 配置） 0 1 推挽 1 0 高阻 1 1 开漏

$\overline{PSEN}$	26	32	程序存储使能：外部程序存储器的读选通。当芯片从外部程序存储器读取程序时，$\overline{PSEN}$ 每个机器周期被激活两次。而在每次访问外部数据存储器时 $\overline{PSEN}$ 被忽略两次。对内部程序程序存储器访问时 $\overline{PSEN}$ 无效(保持为高)。$\overline{PSEN}$ 可驱动 8 个 LSTTL 输入。驱动 CMOS 不需要外部上拉。
ALE/$\overline{PROG}$	27	33	地址锁存使能：正常操作中，在访问外部存储器时锁存地址的低字节。每 6 个振荡器周期激活一次，但在访问外部数据存储器时例外。ALE 可驱动 8 个 LSTTL 输入。驱动 CMOS 不需要外部上拉。若想禁止 ALE 的翻转（降低 RFI 噪声），必须通过软件置位 A0（SFR：AUXR.0）。见表 4 $\overline{PROG}$:编程脉冲输入。
$\overline{EA}$/Vpp	35	29	外部访问输入：如果复位时 $\overline{EA}$ 保持 TTL 高电平，CPU 执行内部程序存储器的程序。如果为 TTL 低电平，CPU 通过 P0 和 P2 执行外部程序存储器的程序。EA 不允许保持悬浮。在复位时锁存，复位后不用考虑。 Vpp：提供编程电压
P0.0/AD0-P0.7/AD7	30-37	36-43	P0 口：8 位开漏双向 I/O 口。复位时 P0 口为高阻态（三态）。 AD7-AD0：复用的数据和地址总线低地址。P0 口可驱动 8 个 LSTTL 输入。
AVref+	38	44	A/D 转换参考电阻：高端
AVss	39	1	模拟地
P1.0-P1.4 P1.5-P1.7	40-44 1-3	2-6 7-9	P1 口：用户可配置输出类型的 8 位 I/O 口。P1 口作为输入或输出时的操作取决于所选择的配置。每个口都可独立配置。 P1 口还提供以下功能：
P1.0 P1.1 P1.5-P1.7 P1.5 P1.6 P1.7	40 41 42-44 1-3 1 2 3	2 3 4-6 7-9 7 8 9	RXDC：CAN 接收器输入脚 TXDC：CAN 发送器输出脚 复位时，P1.0 和 P1.1 异步驱动为高，P1.2-P1.7 为高阻态（三态） CT0I/INT2/CT1I/INT3/CT2I/INT4：T2 捕获定时器输入或外部中断输入。 ADC0-ADC2:可选功能：ADC 输入通道。 ADC3-ADC5：ADC 输入通道 CT3I/INT5：T2 捕获定时器输入或外部中断输入。 SCL：I^2C 串行时钟线。用于 I^2C 时不可使用推挽或准双向模式。 SDA：I^2C 串行数据线。用于 I^2C 时不可使用推挽或准双向模式。 通过 P1M1 和 P1M2 寄存器可将 P1 口设置为 4 种模式之一： P1M1.x　P1M2.x　模式描述 0　0　准双向（默认的标准 C51 配置(2)） 0　1　推挽(2) 1　0　高阻 1　1　开漏
PWM0	6	12	脉宽调制：输出 0
PWM1	28	34	脉宽调制：输出 1

注：1.　为了避免上电时的“闩锁”效应，任意管脚上的电压任何时候都不能高于 VDD+0.5V 或低于 Vss-0.5V。

2. 不可用于 P1.6 和 P1.7。

7 存储器结构

中央处理单元（CPU）在下图所示的 3 个存储空间执行操作数（见图 5）：

- 16K 字节内部程序存储器，可外部扩展到 64K 字节
- 512 字节内部数据存储器，主和辅助 RAM
- 最大 64K 字节外部数据存储器（256 字节位于内部辅助 RAM）

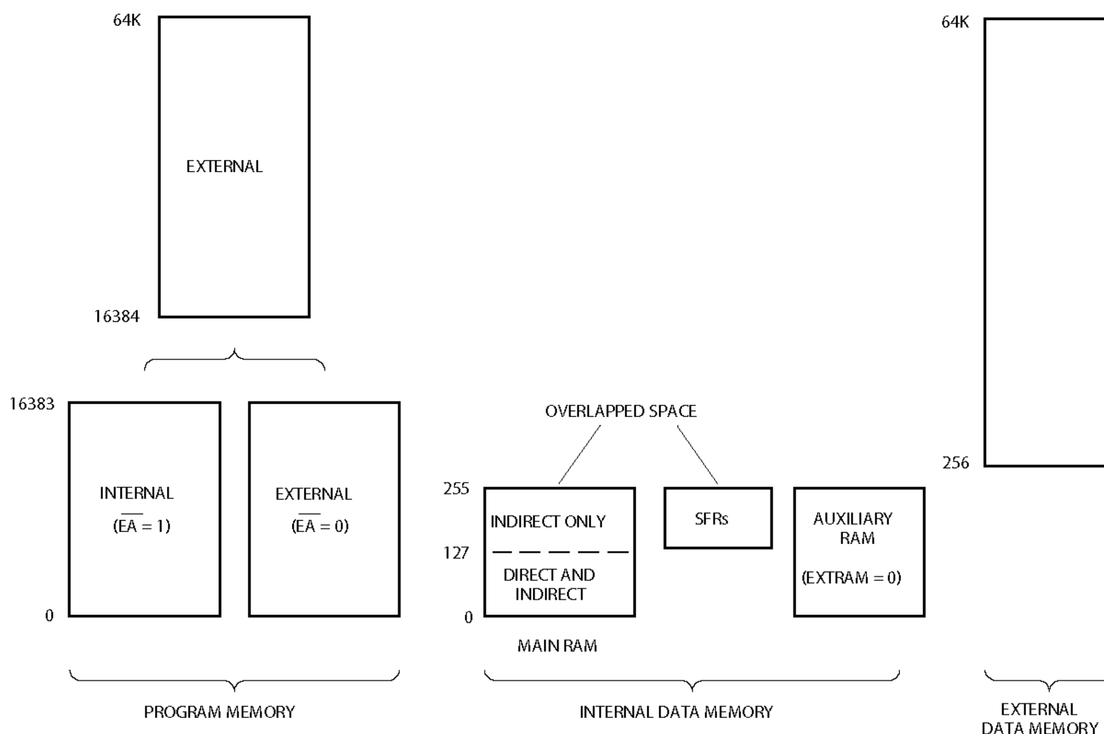


图 5 存储器结构和地址空间（EXTRAM=0）

7.1 程序存储器

PP8xC591 包含 16K 字节片内程序存储器，可使用外部存储器扩展到 64K 字节。当 EA 为高电平时，PP8xC591 从内部 ROM 取指，除非地址超过 3FFFh。地址 4000h 到 FFFFh 取自外部程序存储器。EA 在复位时锁存，复位之后不用考虑。

对于 ROM 和 EPROM 的 PP8xC591，器件执行了防范的措施以保护其不会被非法的程序存储器代码读取。

7.2 寻址

PP8xC591 有 5 种方式用于对程序和数据存储器进行寻址：

- 寄存器
- 直接
- 寄存器间接
- 立即
- 基址寄存器加变址寄存器间接寻址

7.3 扩展的数据 RAM 寻址

PP8xC591 的内部数据存储器分为 4 个独立的部分：低 128 字节 RAM，高 128 字节 RAM，128 字节特殊功能寄存器和 256 字节辅助 RAM（AUX-RAM）。如图 5 所示。

1. 低 128 字节 RAM（地址 00H~7FH）可直接和间接寻址（见图 6）
2. 高 128 字节 RAM（地址 80H~FFH）为间接寻址

3. 特殊功能寄存器，SFR（地址 80H~FFH）只能直接寻址。
4. 256 字节 AUX-RAM（00H~FFH）通过 MOVX 间接寻址（且 EXTRAM 位清零，见表 3）。

低 128 字节可直接或间接寻址。高 128 字节只能间接寻址。高 128 字节占用与 SFR 相同的地址空间，即它们有相同的地址。但在物理上是单独的空间。

当指令访问地址高于 7FH 的内部空间,CPU 通过指令的寻址方式区别对高 128 字节数据 RAM 和 SFR 的访问。

使用直接寻址指令访问 SFR。例如：

```
MOV    0A0H,#data
```

寻址位于 0A0H 的 SFR（为 P2）。

使用间接寻址方式访问高 128 字节 RAM。例如

```
MOV    @R0,#data
```

此处，R0 的内容为 0A0H，是对地址 0A0H 的数据字节进行访问，而不是 P2（地址为 0A0H）。

AUX-RAM 可通过清除 EXTRAM 位，并使用MOVX 指令间接寻址访问。这部份存储器物理上位于片内，逻辑上占用外部数据存储器的头 256 字节。

EXTRAM＝0 时，使用MOVX 指令加上所选的任意一个寄存器组的 R0，R1 或 DPTR 可对 AUX-RAM 间接寻址。对 AUX-RAM 的访问不会影响 P0 口，P3.6(WR#)和 P3.7(RD#)。在外部寻址时P2 SFR 为输出，例如：在 EXTRAM＝0 时，MOVX @R0,#data

R0 的内容为 0A0H，访问地址 0A0H 处的 AUX-RAM，而不是外部存储器。对位置高于 FFH（即 0100H~FFFFH）的外部数据存储器的访问通过MOVX DPTR 指令执行，和标准 80C51 相同。即 P0 和 P2 作为数据/地址总线，P3.6 和 P3.7 作为读和写的时序信号。参见表 4。

在 EXTRAM＝1 时，MOVX @Ri 和 MOVX @DPTR 和标准 80C51 类似。MOVX @Ri 将 P0 口作为 8 位地址和数据复用，其它任意输出管脚可作为输出高位地址，这提供了外部页切换能力。MOVX @DPTR 指令将产生一个 16 位地址。P2 输出高八位地址（DPH 的内容），而 P0 输出复用的低八位地址位（DPL）和数据。MOVX @Ri 和 MOVX @DPTR 在 P3.6（/WR）和 P3.7（/RD）上产生写或读信号。

堆栈指针（SP）可位于内部数据存储器的 256 字节 RAM（低和高 128 字节 RAM）的任何位置，不可位于 AUX-RAM。

表 2 AUX-RAM 页寄存器(地址 8EH)

7	6	5	4	3	2	1	0
—	—	—	—	—	LVADC	EXTRAM	AO

表 3 AUX-RAM 位描述

位	—	功能
7~3	—	保留将来之用[1]
2	LVADC	使能 A/D 低电压操作 LVADC 操作模式 0 关闭 A/D 充电泵 1 打开 A/D 充电泵，要求操作低于 4V
1	EXTRAM	使用 MOVX @RI/@DPTR 访问内部/外部 RAM（00H~FFH） EXTRAM 操作模式 0 使用 MOVX @RI/@DPTR 访问内部 AUX-RAM（00H~FFH） 1 访问外部数据存储器
0	AO	禁止/使能 ALE AO 操作模式 0 允许 ALE 以固定的速率 1/6 振荡器频率输出 1 ALE 仅在执行 VX 或 MOVC 指令时有效

注：

1. **用户软件不应对保留位写入。这些位用于** 80C51 **以后增加的新特征。在这种情况下，新位的复位值或非激活值为零，激活值为** 1**。从保留位读出的值是不确定的。**

2. **复位值为**"xxxxxx10B"

字节地址	(MSB)							(LSB)	十进制
7Fh									127
≈									≈
2Fh	7F	7E	7D	7C	7B	7A	79	78	47
2Eh	77	76	75	74	73	72	71	70	46
2Dh	6F	6E	6D	6C	6B	6A	69	68	45
2Ch	67	66	65	64	63	62	61	60	44
2Bh	5F	5E	5D	5C	5B	5A	59	58	43
2Ah	57	56	55	54	53	52	51	50	42
29h	4F	4E	4D	4C	4B	4A	49	48	41
28h	47	46	45	44	43	42	41	40	40
27h	3F	3E	3D	3C	3B	3A	39	38	39
26h	37	36	35	34	33	32	31	30	38
25h	2F	2E	2D	2C	2B	2A	29	28	37
24h	27	26	25	24	23	22	21	20	36
23h	1F	1E	1D	1C	1B	1A	19	18	35
22h	17	16	15	14	13	12	11	10	34
21h	0F	0E	0D	0C	0B	0A	09	08	33
20h	07	06	05	04	03	02	01	00	32
1Fh–18h	寄存器组 3								31–24
17h–10h	寄存器组 2								23–16
0Fh–08h	寄存器组 1								15–8
07h–00h	寄存器组 0								7–0

图 6 **内部主** RAM **位地址**

7.3.1 特殊功能寄存器

表 4 特殊功能寄存器位地址,符号或可选的口功能

*** = SFR 为可位寻址； # = SFR 从 80C51 修改而来或新增加的**

名称	定义	地址	位功能和位地址								复位值
ACC*	累加器	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
ADCH#	A/D 转换器高	C6H									xxxxxxxxB
ADCON#	A/D 控制	C5H	ADC.1	ADC.0	—	ADCI	ADCS	AADR2	AADR1	AADR0	xx000000B
AUXR#	辅助功能寄存器	8EH	—	—	—	—	—	LVADC	EXTRAM	AO	xxxxx110B
AUXR1#	辅助功能寄存器 1	A2H	ADC8	AIDL	SRST	WDE	WUPD	0	—	DPS	000000x0B
B*	B 寄存器	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
CTCON#	捕获控制寄存器	EBH	CTN3	CNP3	CTN2	CTP2	CTN1	CTP1	CTN0	CTP0	00H
CTH3#	捕获高 3	CFH									xxxxxxxxB
CTH2#	捕获高 2	CEH									xxxxxxxxB
CTH1#	捕获高 1	CDH									xxxxxxxxB
CTH0#	捕获高 0	CCH									xxxxxxxxB
CMH2#	比较高 2	CBH									00H
CMH1#	比较高 1	CAH									00H
CMH0#	比较高 0	C9H									00H
CTL3#	捕获低 3	AFH									xxxxxxxxB
CTL2#	捕获低 2	AEH									xxxxxxxxB
CTL1#	捕获低 1	ADH									xxxxxxxxB
CTL0#	捕获低 0	ACH									xxxxxxxxB
CML2#	比较低 2	ABH									00H
CML1#	比较低 1	AAH									00H
CML0#	比较低 0	A9H									00H
DPTR	数据指针（双字节）										
DPH	指针高字节	83H									00H
DPL	指针低字节	82H									00H
			AF	AE	AD	AC	AA	AB	A9	A8	
IEN0*#	中断使能 0	A8H	EA	EAD	ES1	ES0	ET1	EX1	ET0	EX0	00H
			EF	EE	ED	EC	EB	EA	E9	E8	
IEN1*#	中断使能 1	E8H	ET2	ECAN	ECM1	ECM0	ECT3	ECT2	ECT1	ECT0	00H
			BF	BE	BD	BC	BBB	BA	B9	B8	
IP0*#	中断优先级 0	B8H	—	PAD	PS1	PS0	PT1	PX1	PT0	PX0	x0000000B
			FF	FE	FD	FC	FB	FA	F9	F8	
IP0H#	中断优先级高字节	B7H	—	PADH	PS1H	PS0H	PT1H	PX1H	PT0H	PX0H	x0000000B
IP1*#	中断优先级 1	F8H	PT2	PCAN	PCM1	PCM0	PCT3	PCT2	PCT1	PCT0	00H
IP1H		F7H	PT2H	PCANH	PCM1H	PCM0H	PCT3H	PCT2H	PCT1H	PCT0H	00H
CANMOD	CAN 模式寄存器	C4H									00H
CANCON	CAN 命令(w)和中断(r)	C3H									00H
CANDAT	CAN 数据	C2H									00H
CANADR	CAN 地址	C1H									00H
			C7	C6	C5	C4	C3	C2	C1	C0	

CANSTA*	CAN 状态(r)	C0H	BS	ES	TS	RS	TCS	TBS	DOS	RBS	00H
	CAN 中断使能（w)		BEIE	ALIE	EPIE	WUIE	DOIE	EIE	TIE	RIE	
P1M1	P1 输出模式 1	92H									FCH
P1M2	P1 输出模式 2	93H									00H
P2M1	P2 输出模式 1	94H									00H
P2M2	P2 输出模式 2	95H									00H
P3M1	P3 输出模式 1	9AH									00H
P3M2	P3 输出模式 2	9BH									00H
			B7	B6	B5	B4	B3	B2	B1	B0	
			—	—	CSMR3	CSMR2	CSMR1	CSMR0	RT2	T2	
P3*	P3 口	B0H	RD	WR	T1	T0	INT1	INT0	TxD	RxD	FFH
			A7	A6	A5	A4	A3	A2	A1	A0	
P2*	P2 口	A0H	A15	A14	A13	A12	A11	A10	A9	A8	FFH
			97	96	95	94	93	92	91	90	
			ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	—	—	
P1*	P1 口	90H	SDA	SCL	CT3I	CT2I	CT1I	CT0I	TXDC	RXDC	FFH
			87	86	85	84	83	82	81	80	
P0*	P0 口	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
PCON	电源控制寄存器	87H	SMOD1	SMOD0	POF	WLE	GF1	GF0	PD	IDL	00x00000B
PSW*	程序状态字	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	00H
PWMP#	PWM 预分频器	FEH									
PWMP1#	PWM 寄存器 1	FDH									
PWMP0#	PWM 寄存器 0	FCH									
RTE#	复位使能	EFH					RP35	RP34	RP33	RP32	xxxx0000B
S0ADDR	串口 0 从地址	F9H									00H
S0ADEN	从地址屏蔽	B9H									00H
SP	堆栈指针	81H									07H
S0BUF	串口 0 数据缓冲区	99H									xxxxxxxxB
S0PSL	UART 预分频器值	FAH									00h
S0PSH	UART 预分频器值	FBH	SPS				预分频器高半字节				0xxx0000B
			9F	9E	9D	9C	9B	9A	99	98	
S0CON*	串口 0 控制	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	00H
S1CON#*	串口 1 控制	D8H	CR2	ENS1	STA	ST0	SI	AA	CR1	CR0	00H
S1ADR#	串口 1 地址	DBH	从地址							GC	00H
S1DAT#	串口 1 数据	DAH									00H
S1STA#	串口 1 状态	D9H	SC4	SC3	SC2	SC1	SC0	0	0	0	F8H
			DF	DE	DD	DC	DB	DA	D9	D8	
STE#	设置使能	EEH					SP35	SP34	SP33	SP32	xxxx0000B
TH1	定时器 1 高字节	8DH									00H
TH0	定时器 0 高字节	8CH									00H
TL1	定时器 1 低字节	8BH									00H
TL0	定时器 0 低字节	8AH									00H
TMH2#	定时器 2 高字节	EDH									00H

TML2#	定时器 2 低字节	ECH									00H
TMOD	定时器工作模式	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H
			8F	8E	8D	8C	8B	8A	89	88	
TCON*	定时器 0/1 控制	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
TM2CON#	定时器 2 控制	EAH	T2IS1	T2IS0	T2ER	T2B0	T2P1	T2P0	T2MS1	T2MS0	00H
			CF	CE	CD	CC	CB	CA	C9	C8	
TM2IR#*	定时器 2/CAN 中断标志寄存器	C8H	T2OV	CMI2/CAN	CMI1	CMI0	CTI3	CTI2	CTI1	CTI0	00H
T3#	定时器 3	FFH									00H

7.4 双 DPTR

双 DPTR 结构（如图 7）提供了一种用于寻址外部数据存贮器的方法。有两个 16 位 DPTR 寄存器可以寻址外部存贮器。通过对 AUXR1 中的 DPS 位编程可实现两个 DPTR 寄存器的切换。

当切换 DPTR0 和 DPTR1 时，应当通过软件来保存 DPS。

注意 AURX 的位 2 不能写，而读出值为 0。通过执行 INC DPTR 指令，能对 DPS 快速切换而不会影响其它位。

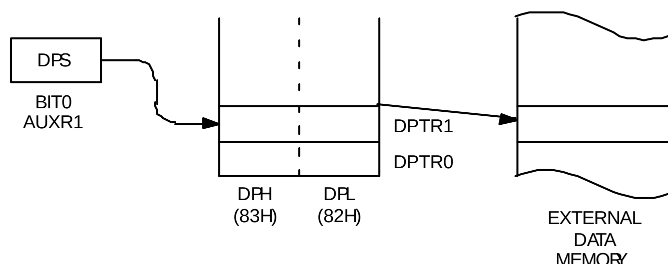


图 7 双 DPTR

DPTR 指令

DPTR 指令根据当前 AUXR1 位 0 的值，可作为数据指针。下面是使用 DPTR 的 6 个指令：

INC	DPTR	数据指针加 1
MOV	DPTR,#data16	DPTR 装载 16 位常数
MOV	A,@A+DPTR	将与 DPTR 相关的代码字节送入 ACC
MOVX	A,@DPTR	外部 RAM（16 位地址）的内容装入 ACC
MOVX	@DPTR,A	把 ACC 的内容送到外部 RAM（16 位地址）
JMP	@A+DPTR	间接跳转到与 DPTR 相关的地址

可以通过寻址 SFR 的低字节或高字节来寻址数据指针。更详细的内容可参见应用指南 AN458。

7.4.1 AUXR1 页寄存器

表 5 AUXR1 页寄存器(地址 A2H)

7	6	5	4	3	2	1	0
ADC8	AIDL	SRST	WDE	WUPD	0	–	DSP

表 6 AUXR1 位描述

用户软件不要将保留位写为 1。这些位可能用于启动将来 8051 家族产品的新特性。那样新功能位的复位或者无效值为逻辑 0，有效值为 1。从保留位读出的值是不确定的。AUXR1 的复位值为（0000000xB）。

位	符号	描述
7	ADC8	ADC 模式切换：10 位和 8 位转换的切换 ADC8 操作模式 0 10 位转换（50 个机器周期） 1 8 位转换（24 个机器周期）
6	AIDL	空闲模式时使能 ADC
5	SRST	软件复位
4	WDE	看门狗定时器使能标志
3	WUPD	使能从掉电模式唤醒
2	0	保留
1	－	保留
0	DSP	数据指针切换：在 DPTR0 和 DPTR1 之间切换 DSP 操作模式 0 DPTR0 1 DPTR1

8 I/O 功能

PP8xC591 包含 32 条 I/O 口线，部分具有复用的功能。I/O 在复位时保持高电平（异步，在振荡器运行之前）。

P0,1,2 和 3 具有下列可选的功能：

P0 与 80C51 相同。复位后口特殊功能寄存器为“FFH”。P0 还提供复用的低位地址和数据总线用于扩展 PP8xC591 的标准存储器和外设。

P1 支持几种可选的功能。因此其具有不同的 I/O 状态。注意，在复位后 P1.0 和 P1.1 为高电平而 P1.2 到 P1.7 为高阻态（三态）。

P2 与 80C51 相同。复位后口特殊功能寄存器为“FFH”。P2 还提供复用的高位地址和数据总线用于扩展 PP8xC591 的外部存储器和/或外部数据存储器。

P3 与 80C51 相同。复位后口特殊功能寄存器为“FFH”。

9 振荡器特性

XTAL1 和 XTAL2 为输入和输出，可分别作为一个反相放大器的输入和输出。此管脚可配置为使用内部振荡器。要使用外部时钟源驱动器件时，XTAL2 可以不连接而由 XTAL1 驱动。外部时钟信号无占空比的要求，因为时钟通过触发器二分频输入到内部时钟电路。但高低电平的最长和最短时间必须符合手册的规定。

10 复位

在振荡器工作时，将 $\overline{RST}$ 脚保持至少两个机器周期高电平（12 个振荡器周期）可实现复位。为了保证上电复位的可靠，RST 保持高电平的时间至少为振荡器启动时间（通常为几个微秒）再加上两个机器周期。

RST 还可被看门狗定时器 T3 所激活的下拉晶体管内部拉低。T3 输出的脉冲宽度为 3 个机器周期。这样一个短的脉宽是必要的，可以尽可能快的使处理器或系统从错误中恢复。

需要注意的是，定时器 T3 输出的短脉冲不能对上电复位电容放电（见图 8）。因此，当看门狗定时器也用于外部器件时，RST 脚不能连接此电容。应当使用一个不同的电路来执行上电复位。

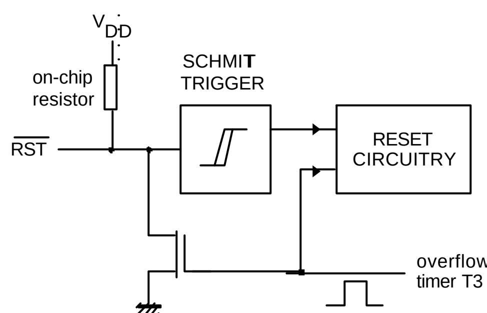


图 8 片内复位配置

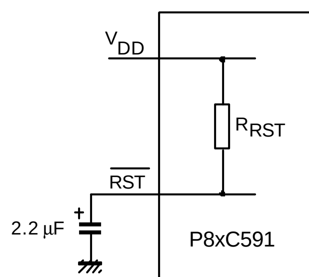


图 9 上电复位

11 低功耗模式

11.1 时钟停止模式

静态设计使时钟频率可以降至 0MHz(停止)。当振荡器停振时，RAM 和 SFR 的值保持不变。该模式允许逐步应用并可将时钟频率降至任意值以实现系统功耗的降低。如要实现最低功耗则建议使用掉电模式。

11.2 空闲模式

空闲模式（见表 2）中，CPU 进入睡眠状态，但片内的外围电路仍然保持工作状态。正常操作模式的最后一条指令执行进入空闲模式。空闲模式下，CPU 内容、片内 RAM 和所有 SFR 保持原来的值。任何被使能的中断（此时，程序从中断服务程序处恢复并继续执行）或硬件复位（与上电复位使用相同的方式启动处理器）均可终止空闲模式。

11.3 掉电模式

为了进一步降低功耗，通过软件可实现掉电模式(见表 2)。该模式中，振荡器停振并且在最后一条指令执行进入掉电模式。降到 2.0V 时，片内 RAM 和 SFR 保持原值，在退出掉电模式之前 Vcc 必须升至规定的最低操作电压。

硬件复位或外部中断均可结束掉电模式。硬件复位使所有的 SFR 重新设置，但不改变片内 RAM 的值。外部中断允许 SFR 和片内 RAM 都保持原值。

WUPD（AUXR1.3－从掉电唤醒）使能或禁止通过外部中断唤醒掉电。

WUPD＝0：禁止　　WUPD=1：使能

要正确退出掉电模式，在 Vcc 恢复到正常操作电压范围之后，复位或外部中断开始执行并且要保持足够长的时间（ 通常小于 10ms)以使振荡器重新启动并稳定下来。

使用外部中断退出掉电模式时，$\overline{\text{INT0}}$ 和 $\overline{\text{INT1}}$ 必须使能且配置为电平触发。将管脚电平拉低使振荡器重新启动，退出掉电模式后将管脚恢复为高电平。一旦中断被响应，RETI 之后所执行的是进入掉电模式指令的后一条指令。

表 2 空闲模式和掉电模式时外部管脚的状态

模式	程序存储器	ALE	$\overline{PSEN}$	口 0	口 1	口 2	口 3	PWM0/PWM1
空闲	内部	1	1	数据	数据	数据	数据	高
	外部	1	1	悬浮	数据	地址	数据	高
掉电	内部	0	0	数据	数据	数据	数据	高
	外部	0	0	悬浮	数据	数据	数据	高

11.3.1 电源关闭标志

当 Vcc 从 0V 上升到 5V 时，片内电路将电源关闭标志（POF）置位。POF 可通过软件置位或清零。这样用户可以确定复位是由上电引起的还是退出掉电模式的热启动。Vcc 必须保持在 3V 以上 POF 才不会受影响。

11.3.2 设计中的注意事项

当空闲模式被硬件复位所中止时，器件在内部复位之前从停止处恢复程序正常运行，时间为 2 个机器周期。这段时间内片内硬件禁止对内部 RAM 的访问，但对 I/O 口的访问未被禁止。当 Idle 模式被复位所中止时，为了消除可能产生的误写操作，应用 Idle 模式指令后的指令不应执行写 I/O 口或写外部存储器操作。

11.3.3 ONCE™ 模式

ONCE(在线仿真)模式实现了对系统的测试和调试而不需要将器件从电路中移去。进入 ONCE 模式的条件：

1． 当器件复位且 $\overline{PSEN}$ 为高电平时，将 ALE 置低电平；
2． 在 RST 撤除时，ALE 保持低电平。

当器件处于 ONCE 模式时，P0 口处于悬浮状态，其它 I/O 口、ALE 和 $\overline{PSEN}$ 为弱上拉。振荡电路保持工作状态，器件处于该模式时，可用仿真器或测试 CPU 驱动电路。执行正常复位时恢复正常操作。

11.3.4 降低 EMI 模式

ALE-Off 位,AO(AUXR.0)可设置为 0 禁止 ALE 输出。当要求对外部存储器进行访问时它自动有效，并在结束对外部存储器的访问之后恢复关闭状态。

11.3.5 电源控制寄存器(PCON)

表 8 电源控制寄存器(地址 87H)

7	6	5	4	3	2	1	0
SMOD1	SMOD0	POF	WLE	GF1	GF0	PD	IDL

表 9 PCON 位描述

如果将 1 同时写入 PD 和 IDL，PD 优先。PCON 的复位值为(0XX00000)。

位	符号	描述
7	SMOD1	波特率加倍:当串口 SIO0 用于模式 1,2 和 3 时,置位该位可使波特率加倍
6	SMOD0	选择 SCON.7 的 SM0/FE,清零时 SCON.7 是 SM0 位,当置位时是 FE(帧错误)标志
5	POF	电源关闭标志
4	WLE	看门狗装载使能：在装载 T3(看门狗)之前必须将该位置位。当 T3 装载后清零
3	GF1	通用标志位
2	GF0	
1	PD	掉电模式选择：该位置位将使能掉电模式，该位只有当看门狗使能位“WDE”设置为零时才能置位。
0	IDL	空闲模式选择：该位置位将使能空闲模式

12 CAN，控制器局域网络

控制器局域网络是一个用于串行通信的高性能通信协议的定义。CAN 控制器的电路是根据 CAN 规范 2.0B 版设计的，提供了对 CAN 协议的完全实现。包含此片内 CAN 控制器的微控制器用于建立通用工业和汽车环境中强大的本地网络。这极大地减少了线束并增强了诊断和监控能力。

PP8xC591 包括了 Philips 半导体的独立 CAN 控制器 SJA1000 所具有的相同功能，并有如下改进：

- 增强的 CAN 接收中断
- 扩展的验收滤波器
  - －8 个滤波器用于标准帧格式
  - －4 个滤波器用于扩展格式
  - －“change on the fly”特性

12.1 PeliCAN 控制器的特性

12.1.1 通用 CAN 特性

- CAN2.0B 协议兼容性
- 多主机结构
- 由信息识别码（11 位或 29 位）确定总线访问优先级
- 非破坏性的位仲裁
- 保障高优先级信息的等待时间
- 可编程传输速率（最大 1Mbit/s）
- 多点传送和广播信息能力
- 数据长度从 0 到 8 个字节
- 强大的错误处理能力
- 带位填充的非归零（NRZ）编码和解码
- 适用于大范围的网络，包括 SAE 网络类别 A,B,C

12.1.2 PP8xC591 PeliCAN 特性（CAN2.0B 新增）

- 同时支持 11 位和 29 位识别码
- 位速率可达 1Mbits/s
- 可读/写访问的错误计数器
- 可编程的错误报警界限
- 具体位位置的错误代码捕获
- 具体位位置的仲裁丢失中断
- 单次发送（无重发）
- 只听模式（无应答，无有效的错误标志）
- 支持热插拔（软件位速率检测）
- 扩展的接收缓冲区（FIFO,64 字节）
- 接收缓冲区级灵敏的接收中断
- 用于接收中断的高优先级验收器滤波器
- 验收器的“change on the fly”特性
- 自身信息接收（自接收请求）
- 可编程的 CAN 输出驱动器配置

12.2 PeliCAN 结构

80C51 CPU 接口将 PeliCAN 与 PP8xC591 微控制器内部总线相连。通过 5 个特殊功能寄存器 CANADR，CANDAT，CANMOD，CANSTA 和 CANCON 对 PeliCAN 进行访问。

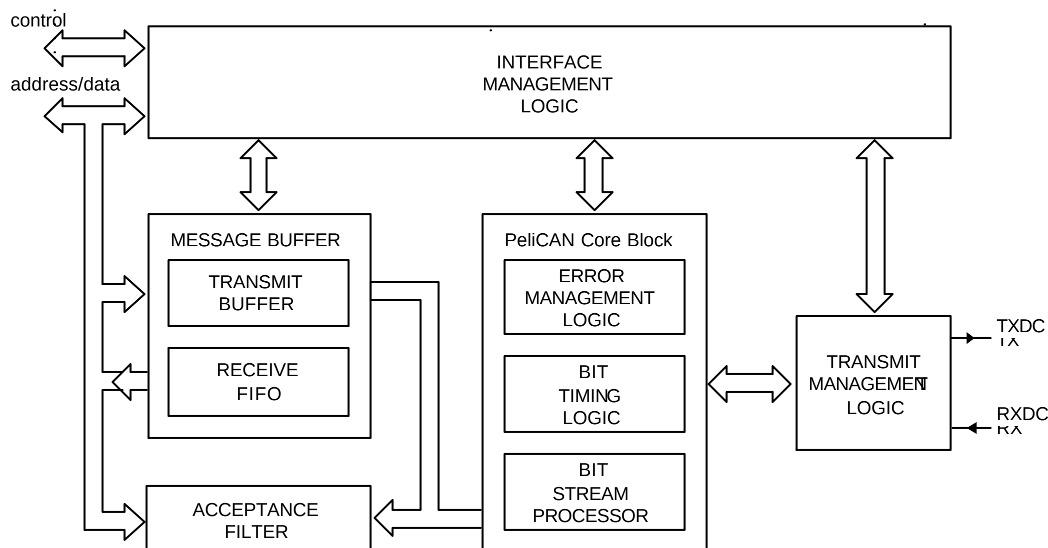


图 10 PeliCAN 方框图

12.2.1 接口管理逻辑（IML）

接口管理逻辑解释来自 CPU 的命令，控制 CAN 寄存器的寻址，向主控制器提供中断信息和状态信息。

12.2.2 发送缓冲器（TXB）

发送缓冲器是 CPU 和 BSP（位流处理器）之间的接口，能够存储发送到 CAN 网络上的完整信息。缓冲器长 13 个字节，由 CPU 写入、BSP 读出。

12.2.3 接收缓冲器（RXB，RXFIFO）

接收缓冲器是验收滤波器和 CPU 之间的接口，用来储存从 CAN 总线上接收和验收的信息。接收缓冲器是一个可被 CPU 访问的接收 FIFO（RXFIFO，长 64 字节）的 13 个字节窗口。

CPU 在此 FIFO 的支持下，可以在处理信息的时候接收其它信息。

12.2.4 验收滤波器（ACF）

验收滤波器把它其中的数据和接收的识别码的内容相比较，以决定是否接收信息。在实际的验收测试中，所有的信息都保存在 RXFIFO 中。

12.2.5 位流处理器（BSP）

位流处理器是一个在发送缓冲器、RXFIFO 和 CAN 总线之间控制数据流的程序装置。它还执行错误检测、仲裁、总线填充和错误处理。

12.2.6 错误管理逻辑（EML）

EML 负责传送层模块的错误管制。它接收 BSP 的出错报告，然后通知 BSP 和 IML 有关的错误统计。

12.2.7 位时序逻辑（BTL）

位时序逻辑监视串口的 CAN 总线和处理与总线有关的位时序。它在信息开头“弱势-支配”的总线传输时同步 CAN 总线位流（硬同步），接收信息时再次同步下一次传送（软同步）。BTL 还提供了可编程的时间段来补偿传播延迟时间、相移（例如，由于振荡漂移）和定义采样点和每一个位时间内的采样次数。

12.2.8 发送管理逻辑（TML）

发送管理逻辑提供驱动器信号用于推挽式的 CAN TX 晶体管级。外部晶体管根据可编程输出驱动器的配置打开或者关闭。此外还执行短路保护和硬件复位的异步悬浮。

12.3 PeliCAN 控制器与 CPU 之间的通信

80C51 CPU 接口将 PeliCAN 与 PP8xC591 微控制器内部总线相连。通过特殊功能寄存器可对 PeliCAN 寄存器和 RAM 区进行便捷的访问。由于支持大范围的地址，基于寻址的间接指针允许使用地址自动增加模式对寄存器进行快速访问。这样就将所需 SFR 的数目减少到 5 个。

- 5 个特殊功能寄存器（SFRs）

- 自动增加模式中的寄存器寻址
- 对 PeliCAN 整个地址范围的访问

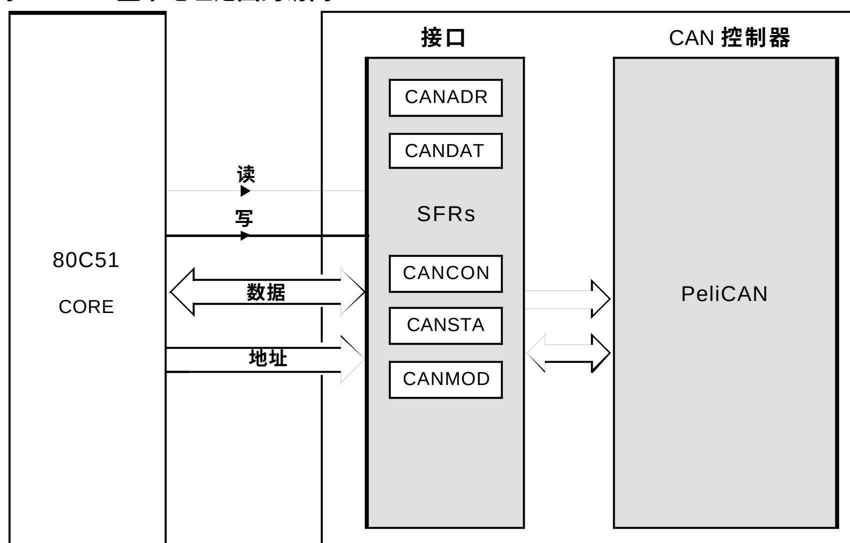


图 11 CPU 与 CAN 的接口

12.3.1 特殊功能寄存器

CPU 通过 5 个特殊功能寄存器 CANADR，CANDAT，CANMOD，CANSTA 和 CANCON 对 PeliCAN 模块进行访问。需要注意的是，CANCON 和 CANSTA 根据访问方向的不同而具有不同的寄存器结构。

PeliCAN 寄存器可以通过两种不同的方式访问。支持软件轮询或控制主要的 CAN 功能的最重要的寄存器象单独的 SFRs 一样直接访问。CAN 模块的其它部分通过一个间接的指针机制进行访问。为了达到高数据吞吐量，在使用间接寻址时也包含了地址自动增加的特性。

表 10 CAN 特殊功能寄存器

SFR	访问	PELICAN REG.	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	SFR 地址
CANADR	读/写	–	CANA7	CANA6	CANA5	CANA4	CANA3	CANA2	CANA1	CANA0	C1
CANDAT	读/写	–	CAND7	CAND6	CAND5	CAND4	CAND3	CAND2	CAND1	CAND0	C2
CANMOD	读/写	模式	TM	RIPM	RPM	SM	–	STM	LOM	RM	C4
CANSTA	读	状态	BS	ES	TS	RS	TCS	TBS	DOS	RBS	C0
	写	中断使能	BEIE	ALIE	EPIE	WUIE	DOIE	EIE	TIE	RIE	
CANCON	读	中断	BEI	ALI	EPI	WUI	DOI	EI	TI	RI	C3
	写	命令	–	–	–	SRR	CDO	RRB	AT	TR	

12.3.2 CANADR

该读/写寄存器定义通过 CANDAT 访问的 PeliCAN 内部寄存器的地址。可将其解释为对 PeliCAN 的一个指针。对 PeliCAN 块寄存器的读/写访问通过 CANDAT 寄存器执行。

通过地址自动增加模式，为 CAN 控制器内部寄存器提供了快速的类栈读和写。如果 CANADR 内当前定义的地址大于或等于 32（十进制），CANADR 的内容在任意对 CANDAT 读或写操作后自动增加。例如，将一个信息装入发送缓冲区可通过将发送缓冲区的首地址（112）写入 CANADR，然后将信息字节一个接一个写入 CANDAT。CANADR 超过 FFH 后复位为 00H。

如果 CANADR 小于 32，不会执行自动地址增加。即使 CANDAT 执行读或写，CANADR 的值仍保持不变。这允许在 PeliCAN 控制器的低地址空间进行寄存器轮询。

13.3.3 CANDAT

CANDAT 作为一个读/写寄存器。特殊功能寄存器 CANDAT 看上去是对 CANADR 所选的 CAN 控制器内部寄存器的一个端口。对 CANDAT 寄存器的读写等效于对该内部寄存器的访问。需要注意的是，如果

CANADR 中当前的地址大于等于 32，那么任何对 CANDAT 的访问将使 CANADR 自动增加。

13.3.4 CANMOD

对 PeliCAN 模式寄存器 CANMOD 是直接进行读写访问的.模式寄存器位于 PeliCAN 模块中的地址 00H。

13.3.5 CANSTA

根据访问方向的不同，CANSTA 提供对 PeliCAN 的状态寄存器和中断使能寄存器的直接访问。对 CANSTA 的读操作是对 PeliCAN 的状态寄存器（地址 2）进行访问。对 CANSTA 的写操作是对中断使能寄存器（地址 4）进行访问。

13.3.6 CANCON

根据访问方向的不同，CANCON 提供对 PeliCAN 的中断寄存器和命令寄存器的直接访问。对 CANCON 的读操作是对 PeliCAN 的中断寄存器（地址 3）进行访问。对 CANCON 的写操作是对命令寄存器（地址 1）进行访问。

12.4 寄存器和信息缓冲区描述

12.4.1 地址分布

PeliCAN 内部寄存器从主 CPU 看来像是一个片内存储器外围寄存器.由于 PeliCAN 可在不同的模式中操作(操作/复位，见模式寄存器)，用户必须在不同的内部寻址定义之间进行区分。从 CAN 地址 128 开始的所有内部 FIFO RAM 都映象到 CPU 接口。

表 11 地址分配

CAN 地址	操作模式		复位模式	
	读	写	读	写
0	模式	模式	模式	模式
1	(00)	命令	(00)	命令
2	状态	—	状态	—
3	中断	—	中断	—
4	中断使能	中断使能	中断使能	中断使能
5	Rx 中断级	Rx 中断级	Rx 中断级	Rx 中断级
6	总线时序 0	—	总线时序 0	—
7	总线时序 1	—	总线时序 1	—
8	见注 2	—	—	—
9	Rx 信息计数器	—	Rx 信息计数器	—
10	Rx 缓冲区起始地址	—	Rx 缓冲区起始地址	—
11	仲裁丢失捕获	—	仲裁丢失捕获	—
12	错误代码捕获	—	错误代码捕获	—
13	错误报警界限	错误报警界限	错误报警界限	错误报警界限
14	Rx 错误计数器	—	Rx 错误计数器	Rx 错误计数器
15	Tx 错误计数器	—	Tx 错误计数器	Tx 错误计数器
16~28	保留(00)	—	保留(00)	—
29	ACF 模式	—	ACF 模式	ACF 模式
30	ACF 使能	ACF 使能	ACF 使能	ACF 使能
31	ACF 优先级	ACF 优先级	ACF 优先级	ACF 优先级

CAN地址		操作模式 读	操作模式 写	复位模式 读	复位模式 写
32	BANK1	验收代码 0	验收代码 0	验收代码 0	验收代码 0
33		验收代码 1	验收代码 1	验收代码 1	验收代码 1
34		验收代码 2	验收代码 2	验收代码 2	验收代码 2
35		验收代码 3	验收代码 3	验收代码 3	验收代码 3
36		验收屏蔽 0	验收屏蔽 0	验收屏蔽 0	验收屏蔽 0
37		验收屏蔽 1	验收屏蔽 1	验收屏蔽 1	验收屏蔽 1
38		验收屏蔽 2	验收屏蔽 2	验收屏蔽 2	验收屏蔽 2
39		验收屏蔽 3	验收屏蔽 3	验收屏蔽 3	验收屏蔽 3
40	BANK2	验收代码 0	验收代码 0	验收代码 0	验收代码 0
41		验收代码 1	验收代码 1	验收代码 1	验收代码 1
42		验收代码 2	验收代码 2	验收代码 2	验收代码 2
43		验收代码 3	验收代码 3	验收代码 3	验收代码 3
44		验收屏蔽 0	验收屏蔽 0	验收屏蔽 0	验收屏蔽 0
45		验收屏蔽 1	验收屏蔽 1	验收屏蔽 1	验收屏蔽 1
46		验收屏蔽 2	验收屏蔽 2	验收屏蔽 2	验收屏蔽 2
47		验收屏蔽 3	验收屏蔽 3	验收屏蔽 3	验收屏蔽 3
48	BANK3	验收代码 0	验收代码 0	验收代码 0	验收代码 0
49		验收代码 1	验收代码 1	验收代码 1	验收代码 1
50		验收代码 2	验收代码 2	验收代码 2	验收代码 2
51		验收代码 3	验收代码 3	验收代码 3	验收代码 3
52		验收屏蔽 0	验收屏蔽 0	验收屏蔽 0	验收屏蔽 0
53		验收屏蔽 1	验收屏蔽 1	验收屏蔽 1	验收屏蔽 1
54		验收屏蔽 2	验收屏蔽 2	验收屏蔽 2	验收屏蔽 2
55		验收屏蔽 3	验收屏蔽 3	验收屏蔽 3	验收屏蔽 3
56	BANK4	验收代码 0	验收代码 0	验收代码 0	验收代码 0
57		验收代码 1	验收代码 1	验收代码 1	验收代码 1
58		验收代码 2	验收代码 2	验收代码 2	验收代码 2
59		验收代码 3	验收代码 3	验收代码 3	验收代码 3
60		验收屏蔽 0	验收屏蔽 0	验收屏蔽 0	验收屏蔽 0
61		验收屏蔽 1	验收屏蔽 1	验收屏蔽 1	验收屏蔽 1
62		验收屏蔽 2	验收屏蔽 2	验收屏蔽 2	验收屏蔽 2
63		验收屏蔽 3	验收屏蔽 3	验收屏蔽 3	验收屏蔽 3

CAN地址	操作模式 读 (SFF)	操作模式 读 (EFF)	操作模式 写	复位模式 读 (SFF)	复位模式 读 (EFF)	复位模式 写 (SFF)	复位模式 写 (EFF)
64~95	保留(00)		—	保留(00)		—	
96	Rx 帧信息	Rx 帧信息		Rx 帧信息	Rx 帧信息	Rx 帧信息	Rx 帧信息
97	Rx 识别码 1	Rx 识别码		Rx 识别码 1	Rx 识别码	Rx 识别码 1	Rx 识别码
98	Rx 识别码 2	Rx 识别码		Rx 识别码 2	Rx 识别码	Rx 识别码 2	Rx 识别码
99	Rx 数据 1	Rx 识别码		Rx 数据 1	Rx 识别码	Rx 数据 1	Rx 识别码
100	Rx 数据 2	Rx 识别码		Rx 数据 2	Rx 识别码	Rx 数据 2	Rx 识别码
101	Rx 数据 3	Rx 数据 1		Rx 数据 3	Rx 数据 1	Rx 数据 3	Rx 数据 1

CAN地址	操作模式				复位模式			
	读		写		读		写	
102	Rx 数据 4	Rx 数据 2			Rx 数据 4	Rx 数据 2	Rx 数据 4	Rx 数据 2
103	Rx 数据 5	Rx 数据 3			Rx 数据 5	Rx 数据 3	Rx 数据 5	Rx 数据 3
104	Rx 数据 6	Rx 数据 4			Rx 数据 6	Rx 数据 4	Rx 数据 6	Rx 数据 4
105	Rx 数据 7	Rx 数据 5			Rx 数据 7	Rx 数据 5	Rx 数据 7	Rx 数据 5
106	Rx 数据 8	Rx 数据 6			Rx 数据 8	Rx 数据 6	Rx 数据 8	Rx 数据 6
107	(FIFO RAM)(1)	Rx 数据 7			(FIFO RAM)(1)	Rx 数据 7	(FIFO RAM)(1)	Rx 数据 7
108	(FIFO RAM)(1)	Rx 数据 8			(FIFO RAM)(1)	Rx 数据 8	(FIFO RAM)(1)	Rx 数据 8
109~111	保留(00)		–		保留(00)		–	
112	(SFF) Rx 帧信息	(EFF) Rx 帧信息	(SFF) Rx 帧信息	(EFF) Rx 帧信息	(SFF) Rx 帧信息	(EFF) Rx 帧信息	(SFF) Rx 帧信息	(EFF) Rx 帧信息
113	Rx 识别码 1	Rx 识别码	Rx 识别码 1	Rx 识别码	Rx 识别码 1	Rx 识别码	Rx 识别码 1	Rx 识别码
114	Rx 识别码 2	Rx 识别码	Rx 识别码 2	Rx 识别码	Rx 识别码 2	Rx 识别码	Rx 识别码 2	Rx 识别码
115	Rx 数据 1	Rx 识别码	Rx 数据 1	Rx 识别码	Rx 数据 1	Rx 识别码	Rx 数据 1	Rx 识别码
116	Rx 数据 2	Rx 识别码	Rx 数据 2	Rx 识别码	Rx 数据 2	Rx 识别码	Rx 数据 2	Rx 识别码
117	Rx 数据 3	Rx 数据 1	Rx 数据 3	Rx 数据 1	Rx 数据 3	Rx 数据 1	Rx 数据 3	Rx 数据 1
118	Rx 数据 4	Rx 数据 2	Rx 数据 4	Rx 数据 2	Rx 数据 4	Rx 数据 2	Rx 数据 4	Rx 数据 2
119	Rx 数据 5	Rx 数据 3	Rx 数据 5	Rx 数据 3	Rx 数据 5	Rx 数据 3	Rx 数据 5	Rx 数据 3
120	Rx 数据 6	Rx 数据 4	Rx 数据 6	Rx 数据 4	Rx 数据 6	Rx 数据 4	Rx 数据 6	Rx 数据 4
121	Rx 数据 7	Rx 数据 5	Rx 数据 7	Rx 数据 5	Rx 数据 7	Rx 数据 5	Rx 数据 7	Rx 数据 5
122	Rx 数据 8	Rx 数据 6	Rx 数据 8	Rx 数据 6	Rx 数据 8	Rx 数据 6	Rx 数据 8	Rx 数据 6
123	(FIFO RAM)(1)	Rx 数据 7	(FIFO RAM)(1)	Rx 数据 7	(FIFO RAM)(1)	Rx 数据 7	(FIFO RAM)(1)	Rx 数据 7
124	(FIFO RAM)(1)	Rx 数据 8	(FIFO RAM)(1)	Rx 数据 8	(FIFO RAM)(1)	Rx 数据 8	(FIFO RAM)(1)	Rx 数据 8
125~127	通用 RAM		通用 RAM		通用 RAM		通用 RAM	
128 … 191	内部 RAM 地址 0(FIFO) … 内部 RAM 地址 63(FIFO)		– – –		内部 RAM 地址 0(FIFO) … 内部 RAM 地址 63(FIFO)		内部 RAM 地址 0(FIFO) … 内部 RAM 地址 63(FIFO)	

注：

1. 这些地址位置反映了当前信息之后的 FIFO RAM 空间。上电后的内容为随机值并包含下一个信息的开始。如果不再收到更多的信息，部分旧的信息可以出现在这。
2. 地址 8 处的寄存器不执行系统功能,而是保留作将来之用。

12.5 CAN 寄存器

12.5.1 复位值

模式寄存器中的复位模式（RM）位的置位将导致中止信息的发送/接收并进入复位模式。复位模式位从 1 到 0 的跳变使 CAN 控制器返回到模式寄存器中定义的模式。

表 12 复位模式配置

“X”表示该寄存器或位的值不受影响。

地址	寄存器	位	符号	名称	由硬件复位	通过软件或由于总线关闭设置 MOD.0
0	模式	MOD.7	TM	测试模式	0(禁止)	0(禁止)
		MOD.6	-	—	X(保留)	X(保留)
		MOD.5	RPM	接收优先级模式	0(低有效)	0(高有效)
		MOD.4	SM	睡眠模式	0(唤醒)	0(唤醒)
		MOD.3	-	—	0(保留)	0(保留)
		MOD.2	STM	自检测模式	0(正常)	X(无变化)
		MOD.1	LOM	只听模式	0(正常)	X(无变化)
		MOD.0	RM	复位模式	1(当前)	1(当前)
1	命令	CMR.7-5	—	—	0(保留)	0(保留)
		CMR.4	SRR	自接收请求	0(空缺)	0(空缺)
		CMR.3	CDO	清除数据溢出	0(无动作)	0(无动作)
		CMR.2	RRB	释放接收缓冲区	0(无动作)	0(无动作)
		CMR.1	AT	中止发送	0(空缺)	0(空缺)
		CMR.0	TR	发送请求	0(空缺)	0(空缺)
2	状态	SR.7	BS	总线状态	0(总线开启)	0(复位)
		SR.6	ES	出错状态	0(ok)	0(复位)
		SR.5	TS	发送状态	1(等待空闲)	0(复位)
		SR.4	RS	接收状态	1(等待空闲)	0(复位)
		SR.3	TCS	发送完毕状态	1(完毕)	0(复位)
		SR.2	TBS	发送缓冲器状态	1(释放)	X(无变化)(1)
		SR.1	DOS	数据溢出状态	0(空缺)	0(复位)
		SR.0	RBS	接收缓冲器状态	0(空)	0(复位)
3	中断	IR.7	BEI	总线错误中断	0(复位)	X(无变化)(1)
		IR.6	ALI	仲裁丢失中断	0(复位)	0(复位)
		IR.5	EPI	错误被动中断	0(复位)	0(复位)
		IR.4	WUI	唤醒中断	0(复位)	0(复位)
		IR.3	DOI	数据溢出中断	0(复位)	0(复位)
		IR.2	EI	错误报警中断	0(复位)	X(无变化)
		IR.1	TI	发送中断	0(复位)	0(复位)
		IR.0	RI	接收中断	0(复位)	0(复位)
4	中断使能	IER.7	BEIE	总线错误中断使能	X(无变化)	X(无变化)
		IER.6	ALIE	仲裁丢失中断使能	X(无变化)	X(无变化)
		IER.5	EPIE	错误被动中断使能	X(无变化)	X(无变化)
		IER.4	WUIE	唤醒中断使能	X(无变化)	X(无变化)
		IER.3	DOIE	数据溢出中断使能	X(无变化)	X(无变化)
		IER.2	EIE	错误报警中断使能	X(无变化)	X(无变化)
		IER.1	TIE	发送中断使能	X(无变化)	X(无变化)
		IER.0	RIE	接收中断使能	X(无变化)	X(无变化)

地址	寄存器	位	符号	名称	由硬件复位	通过软件或由于总线关闭设置 MOD.0
5	Rx 中断级	—	RIL	Rx 中断级	00000000b	X(无变化)
6	总线定时 0	BTR0.7	SJW.1	同步跳转宽度1	X(无变化)	X(无变化)
		BTR0.6	SJW.0	同步跳转宽度0	X(无变化)	X(无变化)
		BTR0.5	BRP.5	波特率预设值5	X(无变化)	X(无变化)
		BTR0.4	BRP.4	波特率预设值4	X(无变化)	X(无变化)
		BTR0.3	BRP.3	波特率预设值3	X(无变化)	X(无变化)
		BTR0.2	BRP.2	波特率预设值2	X(无变化)	X(无变化)
		BTR0.1	BRP.1	波特率预设值1		
		BTR0.0	BRP.0	波特率预设值0	X(无变化)	X(无变化)
7	总线定时 1	BTR1.7	SAM	采样	X(无变化)	X(无变化)
		BTR1.6	TSEG2.2	时间段2.2	X(无变化)	X(无变化)
		BTR1.5	TSEG2.1	时间段2.1	X(无变化)	X(无变化)
		BTR1.4	TSEG2.0	时间段2.0	X(无变化)	X(无变化)
		BTR1.3	TSEG1.3	时间段1.3	X(无变化)	X(无变化)
		BTR1.2	TSEG1.2	时间段1.2	X(无变化)	X(无变化)
		BTR1.1	TSEG1.1	时间段1.1	X(无变化)	X(无变化)
		BTR1.0	TSEG1.0	时间段1.0	X(无变化)	X(无变化)
9	Rx 信息计数器	—	RMC	Rx 信息计数器	0	0
10	Rx缓冲区起始地址	—	RBSA	Rx 缓冲区起始地址	00000000b	X(无变化)
11	总裁丢失捕获	—	ALC	仲裁丢失捕获	0	X(无变化)
12	错误代码捕获	—	ECC	错误代码捕获	0	X(无变化)
13	错误报警界限	—	EWLR	错误报警界限	96d	X(无变化)
14	Rx 错误计数器	—	RXERR	接收错误计数器	0(复位)	X(无变化) (2)
15	Tx 错误计数器	—	TXERR	发送错误计数器	0(复位)	X(无变化) (2)
29	ACF 模式	ACFMOD.7	MFORMATB4	信息格式区 4	0(SFF)	X(无变化)
		ACFMOD.6	AMODEB4	验收滤波器模式区4	0(双)	X(无变化)
		ACFMOD.5	MFORMATB3	信息格式区3	0(SFF)	X(无变化)
		ACFMOD.4	AMODEB3	验收滤波器模式区3	0(双)	X(无变化)
		ACFMOD.3	MFORMATB2	信息格式区2	0(SFF)	X(无变化)
		ACFMOD.2	AMODEB2	验收滤波器模式区2	0(双)	X(无变化)
		ACFMOD.1	MFORMATB1	信息格式区1	0(SFF)	X(无变化)
		ACFMOD.0	AMODEB1	验收滤波器模式区1	0(双)	X(无变化)
30	ACF 使能	ACFEN.7	B4F2EN	Bank4 滤波器 2 使能	X(无变化)	X(无变化)
		ACFEN.6	B4F1EN	Bank4滤波器1使能	X(无变化)	X(无变化)
		ACFEN.5	B3F2EN	Bank3滤波器2使能	X(无变化)	X(无变化)
		ACFEN.4	B3F1EN	Bank3滤波器1使能	X(无变化)	X(无变化)
		ACFEN.3	B2F2EN	Bank2滤波器2使能	X(无变化)	X(无变化)
		ACFEN.2	B2F1EN	Bank2滤波器1使能	X(无变化)	X(无变化)
		ACFEN.1	B1F2EN	Bank1滤波器2使能	X(无变化)	X(无变化)
		ACFEN.0	B1F1EN	Bank1滤波器1使能	X(无变化)	X(无变化)

地址	寄存器		位	符号	名称	由硬件复位	通过软件或由于总线关闭设置 MOD.0
31	ACF 模式		ACFPRIO.7	B4F2PRIO	Bank4 滤波器 2 优先级	X(无变化)	X(无变化)
			ACFPRIO.6	B4F1PRIO	Bank4滤波器1优先级	X(无变化)	X(无变化)
			ACFPRIO.5	B3F2PRIO	Bank3滤波器2优先级	X(无变化)	X(无变化)
			ACFPRIO.4	B3F1PRIO	Bank3滤波器1优先级	X(无变化)	X(无变化)
			ACFPRIO.3	B2F2PRIO	Bank2滤波器2优先级	X(无变化)	X(无变化)
			ACFPRIO.2	B2F1PRIO	Bank2滤波器1优先级	X(无变化)	X(无变化)
			ACFPRIO.1	B1F2PRIO	Bank1滤波器2优先级	X(无变化)	X(无变化)
			ACFPRIO.0	B1F1PRIO	Bank1滤波器1优先级	X(无变化)	X(无变化)
32~35	Bank1	ACR0~3	—	ACR0~ACR3	验收代码寄存器	X(无变化)	X(无变化)
36~39		AMR0~3	—	AMR0~AMR3	验收屏蔽寄存器	X(无变化)	X(无变化)
40~43	Bank2	ACR0~3	—	ACR0~ACR3	验收代码寄存器	X(无变化)	X(无变化)
44~47		AMR0~3	—	AMR0~AMR3	验收屏蔽寄存器	X(无变化)	X(无变化)
48~51	Bank3	ACR0~3	—	ACR0~ACR3	验收代码寄存器	X(无变化)	X(无变化)
52~55		AMR0~3	—	AMR0~AMR3	验收屏蔽寄存器	X(无变化)	X(无变化)
56~59	Bank4	ACR0~3	—	ACR0~ACR3	验收代码寄存器	X(无变化)	X(无变化)
60~63		AMR0~3	—	AMR0~AMR3	验收屏蔽寄存器	X(无变化)	X(无变化)
96~108	Rx 缓冲区		—	RXB	接收缓冲区	X(空) (3)	X(空) (3)
112~124	Tx 缓冲区		—	TXB	发送缓冲区	X(无变化)	X(无变化)
125~127	通用 RAM		—	—	通用RAM	X(无变化)	X(无变化)

注：

1. 总线关闭时错误报警中断位被置位(此中断使能的情况下)。
2. 如果是因为总线关闭而进入复位模式，接收错误计数器清零，同时发送错误计数器初始化为 127 对 CAN 定义的总线关闭恢复时间（其中包括 11 个连续隐性位的 128 次出现）倒计数。
3. RXFIFO 的内部读/写指针复位为初始值。随后对 RXB 的读操作将得到未定义的数据值（部分旧信息）。发送信息时，信息并行写入接收缓冲器，如果该次发送是由自接收请求强制产生的，将只产生接收中断。所以，即使接收缓冲器是空的，最近一次发送的信息也可从接收缓冲器读出，直到它被下一条发送或接收的信息取代。硬件复位时，RXFIFO 的指针复位到 RAM 的物理地址‘0’。软件设置 MOD.0 或由于总线的关闭，RXFIFO 的指针将被复位到当前有效 FIFO 的起始地址（RBSA 寄存器），这个地址不同于第一次发布接收缓冲器命令后的物理 RAM 地址‘0’。

12.5.2 模式寄存器（MOD）

模式寄存器的内容用于改变 CAN 控制器的状态。其中的位可由 CPU 置位或复位。保留位读出为 0。

表 13 模式寄存器（MOD）位描述说明 CAN 地址 0

位	符号	名称	值	功能
MOD.7	TM	测试模式[1]	1(启动)	在系统时钟的上升沿时，TXDC 脚将映射 RXDC 脚检测到的位。RPM 位在此模式中不起作用
			0(禁止)	
MOD.6	RIPM	保留	—	—
MOD.5	RPM	接收优先级模式	1(高有效)	RXD 输入高有效
			0(低有效)	RXD 输入低有效
MOD.4	SM	睡眠模式[2]	1(高有效)	如果没有等待处理的 CAN 中断且总线无活动，CAN 控制器进入睡眠模式
			0(低有效)	
MOD.3	—	保留	—	—
MOD.2	STM	自测模式[1]	1(自测试)	在该模式中使用自接收请求命令可对全部节点进行测试而不需要总线上任何其它有效的节点。CAN 控制器将执行一个成功的发送，即使没有接收到应答。
			0(正常)	成功发送需要一个应答
MOD.1	LOM	只听模式[3]	1(复位)	在该模式中，CAN 不会对 CAN 总线产生应答，即使成功的接收了一个信息。不会对总线产生有效的错误标志。错误计数器停止在当前值。
			0(正常)	正常通信
MOD.0	RM	复位模式[4]	1(复位)	置位复位模式位将导致当前信息发送/接收的中止并进入复位模式。
			0(正常)	复位模式位的负跳变使 CAN 控制器返回操作模式。

注：

1. 如果之前进入了复位模式，那么只能对 MOD.1，MOD.2，MOD.5，MOD.6 和 MOD.7 进行写操作。
2. 如果睡眠模式位置为“1”并且没有总线的活动和等待处理的中断，PeliCAN 模块将进入睡眠模式。SM 的置位加上之前两种例外情况至少其中之一有效时将会导致产生唤醒中断。SM 电平设为低（唤醒）或有总线活动时，CAN 控制器将会唤醒。唤醒时产生一个唤醒中断。由于总线活动唤醒的直到检测到 11 个连续的隐藏位（总线空闲序列）后才能接收这条信息。注意在复位模式中是不能设置 SM 的。清除复位模式后，再一次检测到总线空闲时，SM 的设置才开始有效。
3. 该操作模式使 CAN 控制器进入错误消极状态。不能进行信息传送。以软件驱动的位速检测和“热插拔”时可使用只听模式。所有其它功能都能象在正常工作模式中一样使用。
4. 在硬件复位或总线状态位为 1（总线关闭）时，复位模式位也被置为 1（当前）。复位模式位为 0 后，CAN 控制器会等待：
   a) 一个总线空闲信号（11 个隐藏位），如果上一次复位是硬件复位或 CPU 初始复位。
   b) 128 个总线空闲，如果上一次复位是 CAN 控制器在重新进入总线开启之前初始化复位。

12.5.3 命令寄存器（CMR）

命令寄存器的内容用于改变 CAN 控制器的状态。控制位可由 CPU 置位或复位，CPU 将命令寄存器看作一个只写存储器。

表 14 命令寄存器（CMR）各位的功能说明；CAN 地址 1

位	符号	名称	值	功能
CMR.7~5	—	保留	—	—
CMR.4	SRR	自接收请求(1) (6)	1(当前) 0(空缺)	信息将被同时发送和接收
CMR.3	CDO	清除数据溢出(2)	1(清除) 0(无动作)	清除数据溢出状态位
CMR.2	RRB	释放接收缓冲器(3)	1(释放) 0(无动作)	接收缓冲器,RXFIFO中当前的信息存储空间将被释放 无动作
CMR.1	AT	中止发送(4) (6)	1(当前) 0(空缺)	如果不是在处理过程中,等待处理的发送请求将被取消
CMR.0	TR	发送请求(5) (6)	1(当前) 0(空缺)	信息将被发送

注：

1. 自接收请求时，如果验收滤波器设置为相应的识别码，信息被同时发送和接收。接收和发送中断将指示正确的自接收。
2. 该命令位用于清除由数据溢出状态位通知的数据溢出状况。数据溢出状态位置位后将不再产生数据溢出中断。
3. 在读取接收缓冲区内容之后，CPU 可通过将 RRB 位置位释放 RXFIFO 存储空间。这样会使接收缓冲器内的另一条信息立即有效。如果没有其它有用的信息，接收中断位复位。如果在 FIFO 中没有高优先级的信息(见验收滤波器描述)并且有效的信息字节等于或小于接收中断级寄存器所定义的值，接收中断也复位。如果执行 RRB 命令，在产生新的接收中断和更新 Rx 缓冲区起始地址之前至少经过 2 个内部时钟周期。
4. 当 CPU 需要推迟之前请求的发送时使用 AT 位。例如：先发送一条比较紧急的信息。当前正在处理的传送不会停止。要想知道原始信息是否成功发送，可以检查传送完毕状态位。这应当在发送缓冲器状态位置 1 或产生发送中断后执行。
5. 如果前一条指令中发送请求位或自接收请求位置“1”，不能通过复位该位来取消，所请求的发送只能通过置位 AT 位中止。
6. 同时置位命令位 CMR.0 和 CM.1 会产生一次信息发送。当发生错误或仲裁丢失时不会重发(单次发送)。同时置位命令位 CMR.4 和 CMR.1 会立即产生一次自接收性质的信息发送。发生错误或仲裁丢失时不会重发。同时置位命令位 CMR.0、CMR.1 和 CMR.4 会立即产生一个信息发送（见 CMR.0 和 CMR.1 的定义）。一旦状态寄存器的发送状态位被置位，内部发送请求就被自动清零。同时置位 CMR.0 和 CMR.4 会忽略 CMR.4 位。

12.5.4 状态寄存器（SR）

状态寄存器的内容反映了 CAN 控制器的状态。状态寄存器对微控制器来说是一个只读存储器。

表 15 状态寄存器（SR）的位功能说明；CAN 地址 2

位	符号	名称	值	功能
SR.7	BS	总线状态；注1	1(总线关闭) 0(总线开启)	CAN控制器不参与总线活动 CAN控制器参与总线活动
SR.6	ES	错误状态；注2	1(错误) 0(ok)	至少一个错误计数器达到或超过了由错误报警限制寄存器(EWLR)定义的CPU报警限制 两个错误计数器都在报警限制以下
SR.5	TS	发送状态；注3	1(发送) 0(空闲)	CAN控制器正在发送信息
SR.4	RS	接收状态；注3	1(接收) 0(空闲)	CAN控制器正在接收信息
SR.3	TCS	发送完成状态；注4	1(完成) 0(未完成)	上一次请求的发送已成功完成 前一次请求的发送未完成
SR.2	TBS	发送缓冲器状态；注5	1(释放) 0（锁定）	CPU可以向发送缓冲器中写入信息 CPU不能访问发送缓冲器。信息正在等待发送或正在发送过程中。
SR.1	DOS	数据溢出状态；注6	1(溢出) 0(空缺)	信息因RXFIFO中无足够的存储空间而丢失 自从上一次执行清除数据溢出命令以来没有数据溢出发生
SR.0	RBS	接收缓冲器状态；注7	1(满) 0(空)	RXFIFO中有一个或多个完整的可用信息 无可用信息

注：

1. 当发送错误计数器超过限制（255），总线状态位被置为 1（总线关闭），CAN 控制器将设置复位模式位为 1（当前）并产生一个错误报警和总线错误中断（相应的中断使能）。发送错误计数器被设置为 127。该模式将会一直保持到 CPU 将复位模式位清零。完成之后，CAN 控制器将通过发送错误计数器的递减计数以等待协议规定的最少时间（128 个总线空闲信号）。之后总线状态位被清零（总线开启），错误状态位被置为 0（ok），错误计数器复位并产生一个错误中断（如果中断使能）。这期间读 TX 错误计数器将给出关于总线关闭恢复的状态信息。
2. 根据 CAN 协议规定，在接收和发送期间检测到错误会影响错误计数器。至少一个错误计数器达到或超过 CPU 报警限制（EWLR）时错误状态位被置位。如果中断使能，会产生错误报警中断。EWLR 硬件复位后的默认值是 96。
3. 如果接收状态位和发送状态位都是 0（空闲），则 CAN 总线是空闲的。
4. 一旦发送请求位或自接收请求位被置 1，发送完成状态位就会置 0（未完成）。发送完成状态位会保持为 0 直到发送成功。
5. 如果 CPU 试图在发送缓冲器状态位是 0（锁定）时对发送缓冲器进行写操作，写入的字节将不被接受且在没有任何提示的情况下丢失。
6. 当要接收的信息已经成功通过验收滤波器的时候，CAN 控制器需要在 RXFIFO 中有足够的空间来存储信息描述符和每一个接收的数据字节。如果没有足够的空间来存储信息，信息就会丢失，在信息变为有效时向 CPU 指示数据溢出。如果信息没有被成功接收（例如，由于错误），就不会指示数据溢出。
7. 读出 RXFIFO 中的所有信息并用释放接收缓冲区命令释放它们的存储空间之后，此位被清零。

12.5.5 中断寄存器（IR）

中断寄存器允许中断源的识别。当这个寄存器的一位或多位被置位时，CAN 中断将反映到 CPU 。CPU 读此寄存器的时候，除了接收中断外的所有位都被复位。

中断寄存器对 CUP 来说是只读存储器。

表 16 中断寄存器（IR）的位功能说明；CAN 地址 3

位	符号	名称	值	功能
IR.7	BEI	总线错误中断	1(置位) 0(复位)	当CAN控制器检测到总线错误且中断使能寄存器中的BEIE置位时，该位置位
IR.6	ALI	仲裁丢失中断	1(置位) 0(复位)	当CAN控制器丢失仲裁并变为接收器且中断使能寄存器中的ALIE为被置位时，此位被置位。在仲裁丢失中断事件之后，该中断被锁定直到错误代码捕获寄存器被读出。
IR.5	EPI	错误消极中断	1(置位) 0(复位)	当CAN控制器到达错误消极状态(至少一个错误计数器超过协议规定的值127)或从错误消极状态又进入错误活动状态以及中断寄存器的EPIE位被置位时此位被置1
IR.4	WUI	唤醒中断；注1	1(置位) 0(复位)	当CAN控制器在睡眠模式中，检测到总线的活动且中断寄存器的WUIE位被置1时此位被置位
IR.3	DOI	数据溢出中断	1(置位) 0(复位)	数据溢出状态位的‘0-1’跳变且中断寄存器的DOIE位被置位时此位被置位
IR.2	EI	错误报警中断	1(置位) 0(复位)	错误状态位或总线状态位的改变（置位和清零）和中断寄存器的EIE位被置位时此位被置位
IR.1	TI	发送中断；注2	1(置位) 0(复位)	发送缓冲器状态从’0-1’(释放)跳变且中断寄存器的TIE位被置位时此位被置位
IR.0	RI	接收中断；注2	1(置位) 0(复位)	接收FIFO不空且中断寄存器的RIE位被置位时此位被置位

注：

1. 如果 CPU 在 CAN 控制器参与总线活动或 CAN 中断正在等待时试图置位睡眠模式位也产生唤醒中断。
2. 除了 RI 取决于相应的中断使能位（RIE）这一点外，此位的行为和接收缓冲器状态位是等效的。所以读中断寄存器时接收中断位不被清除。释放接收缓冲器的命令可以暂时清除 RI。如果执行释放命令后 FIFO 中还有可用信息，RI 被重新置位。否则 RI 保持清 0 状态。

12.5.6 中断使能寄存器（IER）

该寄存器允许使能不同类型的中断源通知 CPU。该寄存器对 CPU 来说是可读/写存储器。

表 17 中断使能寄存器（IER）的各位的功能说明；CAN 地址 4

位	符号	名称	值	功能
IER.7	BEIE	总线错误中断使能	1(使能)	如果检测到总线错误,则CAN控制器请求相应的中断
			0(禁止)	
IER.6	ALIE	仲裁丢失中断使能	1(使能)	如果CAN控制器已丢失了仲裁,则请求相应的中断
			0(禁止)	
IER.5	EPIE	错误消极中断使能	1(使能)	若CAN控制器的错误状态改变(从消极到活动或反之)，则请求相应的中断
			0(禁止)	
IER.4	WUIE	唤醒中断使能	1(使能)	如果睡眠模式中的CAN控制器被唤醒，则请求相应的中断
			0(禁止)	
IER.3	DOIE	数据溢出中断使能	1(使能)	如果数据溢出状态位置位(见状态寄存器),CAN控制器请求相应的中断
			0(禁止)	
IER.2	EIE	错误报警中断使能	1(使能)	如果错误或总线状态改变(见状态寄存器),CAN控制器请求相应的中断
			0(禁止)	
IER.1	TIE	发送中断使能	1(使能)	当信息成功发送或发送缓冲器可再次访问(例如,中止发送命令后)时,CAN控制器请求相应的中断
			0(禁止)	
IER.0	RIE	接收中断使能	1(使能)	当接收缓冲器状态为’满’时,CAN控制器请求相应的中断
			0(禁止)	

12.5.7 Rx 中断级寄存器（RIL）

RIL 寄存器用于定义 RXFIFO 的接收中断级。如果 RXFIFO 中的有效 CAN 信息字节的数目超过了该寄存器定义的值，将产生一个接收中断。注意在完整的信息被接收之后才能产生接收中断。如果 RIL 设置为 00，PeliCAN 功能类似于 SJA1000 的接收中断状态。

表 18 Rx 中断级（RIL）的位说明

CAN 地址.5		Rx 中断级(RIL)					
7	6	5	4	3	2	1	0
RIL.7	RIL.6	RIL.5	RIL.4	RIL.3	RIL.2	RIL.1	RIL.0

12.5.8 总线定时寄存器 0（BTR0）

7	6	5	4	3	2	1	0
SJW.1	SJW.0	BRP.5	BRP.4	BRP.3	BRP.2	BRP.1	BRP.0

12.5.8.1 波特率预分频器（BRP）

CAN 系统时钟 t_{scl} 的周期可编程并决定了单独的位定时。CAN 系统时钟通过下面的等式进行计算：

12.5.8.2 同步跳转宽度（SJW）

为了补偿不同总线控制器的时钟振荡器之间的相移，任何总线控制器都必须在当前发送的任意相关信号边沿重新同步。同步跳转宽度定义了一个位周期可被一个重新同步缩短或延长的最大时钟周期数。

12.5.9 总线定时寄存器 1（BTR1）

总线定时寄存器 1 定义位周期的长度、采样点的位置和每个位时间的采样数目。如果复位模式有效，寄存器可被访问（读/写）。在操作模式中该寄存器只读。

表 20 总线定时寄存器 1（BTR1）（CAN 地址 7）

7	6	5	4	3	2	1	0
SAM	TSEG2.2	TSEG2.1	TSEG2.0	TSEG1.3	TSEG1.2	TSEG1.1	TSEG1.0

12.5.9.1 采样（SAM）

表 21 采样（SAM）

位	值	功能
SAM	1	总线被采样 3 次 —推荐用于低/中高速总线（A 和 B 类）
	0	总线被采样 1 次 —推荐用于高速总线（SAE C 类）

12.5.9.2 时间段 1（TSEG1）和时间段 2（TSEG2）

TSEG1 和 TSEG2 决定每个位周期的时钟数目和采样点的位置：

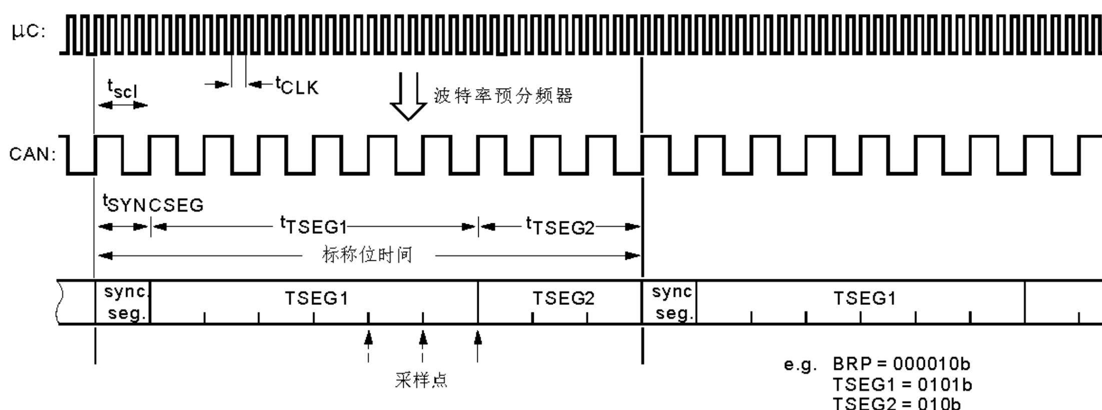


图 12 位周期通常结构

12.5.10 RX 信息计数器（RMC）

RMC 寄存器寄存器（CAN 地址 9）反映了在 RXFIFO 中有效信息的数目。在每次接收事件后增加而由释放接收缓冲区命令减少。任何复位事件后该寄存器清零。

表 22 RX 信息计数器（RMC）（CAN 地址 9）

7	6	5	4	3	2	1	0
RMC.7	RMC.6	RMC.5	RMC.4	RMC.3	RMC.2	RMC.1	RMC.0

12.5.11 RX 缓冲区起始地址（RBSA）

RBSA 寄存器（CAN 地址 10）反映了当前有效的内部 RAM 地址。该地址存放了由接收缓冲区窗口分配的接收信息的第一个字节.这样就有可能说明内部 RAM 的内容。内部 RAM 区从 CAN 地址 32 起始，可由 CPU 进行读/写访问（仅在复位模式中写）。

例：如果 RBSA 设置为 24（十进制），当前在接收缓冲区窗口（CAN 地址 96-108）可见的信息保存在以地址 24 起始的内部 RAM。由于 RAM 直接被分配在 CAN 地址 128（等效于 RAM 地址 0）起始的空间，该信息也可从 CAN 地址 152 以及后面的字节（CAN 地址＝RBSA+128＝24+128＝152）进行访问。

当在 FIFO 内有一个以上的信息时总是执行释放接收缓冲区命令。RBSA 更新为下一个信息的起始地址。上电复位时，指针初始化为 00h。软件复位时（设置复位模式）该指针保持原来的值，但 FIFO 清零，这意味着 RAM 的内容没有改变但下个接收（或发送）的信息将覆盖当前在接收缓冲区窗口中可见的信息。

RX 缓冲区起始地址寄存器对 CPU 来说在操作模式中为只读存储器而在复位模式中为读/写存储器。

表 23 RX 缓冲区起始地址（RBSA）（CAN 地址 10）

7	6	5	4	3	2	1	0
RBSA.7	RBSA.6	RBSA.5	RBSA.4	RBSA.3	RBSA.2	RBSA.1	RBSA.0

12.5.12 仲裁丢失捕获寄存器（ALC）

这个寄存器包括了仲裁丢失的位置的信息。仲裁丢失捕捉寄存器对 CPU 来说是只读存储器。保留位的读值为 0。

表 24 仲裁丢失捕获（ALC）（CAN 地址 11）

7	6	5	4	3	2	1	0
—	—	—	BITNO4	BITNO3	BITNO2	BITNO1	BITNO0

表 25 仲裁丢失捕获寄存器（ALC）的位说明；CAN 地址 11

位	符号	名称	值	功能
ALC.7-ALC.5	—	—	—	保留
ALC.4	BITNO4	第4位	二进制编码帧仲裁丢失位编号 00－仲裁丢失在识别码的第1位	
ALC.3	BITNO3	第3位	… 11－仲裁丢失在SRTR位（RTR位用于标准帧信息）	
ALC.2	BITNO2	第2位	12－仲裁丢失在IDE位 13－仲裁丢失在识别码的第12位（仅扩展帧）	
ALC.1	BITNO1	第1位	…	
ALC.0	BITNO0	第0位	30－仲裁丢失在识别码的最后1位（仅扩展帧） 31－仲裁丢失RTR位（仅扩展帧）	

仲裁丢失时，会产生相应的仲裁丢失中断（如果中断使能）。同时，位流处理器当前位的位置被捕获到仲裁丢失捕获寄存器。用户通过软件读出其内容之前，寄存器中的内容都固定不变。随后，捕获机制被再次激活。读中断寄存器时，中断寄存器中相应的中断标志位被清除。直到仲裁丢失捕捉寄存器被读一次之后，新的仲裁丢失中断才有效。

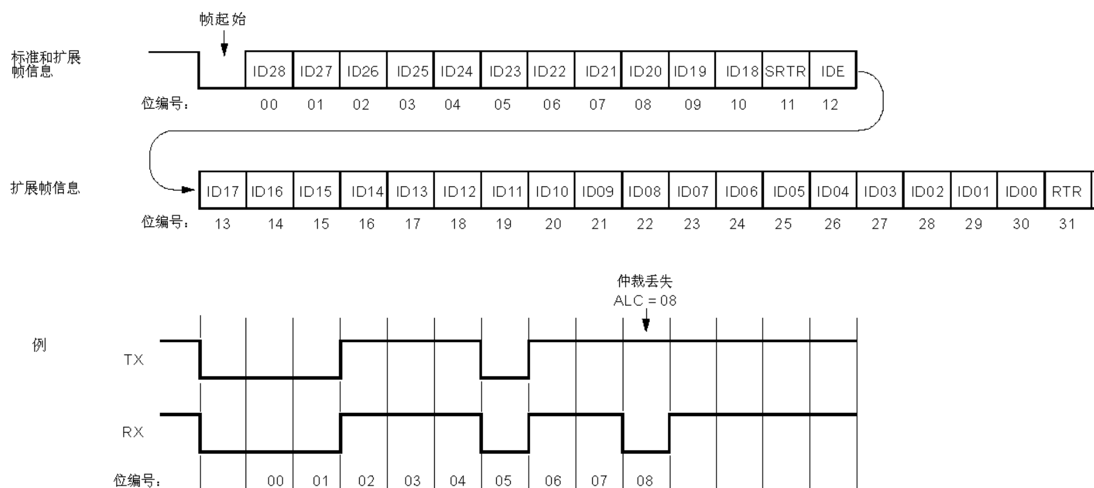


图 13 仲裁丢失位说明

12.5.13 错误代码捕获（ECC）

该寄存器包含了总线错误类型和位置的信息。错误代码捕获寄存器对 CPU 来说是一个只读存储器。

表 26 错误代码捕获（ECC）（CAN 地址 12）

7	6	5	4	3	2	1	0
ERRC1	ERRC0	DIR	SEG4	SEG3	SEG2	SEG1	SEG0

表 27 错误代码捕获寄存器的位说明；CAN 地址 12

位	符号	名称	值		功能
7	ERRC1	错误代码1	ERRC1	ERRC0	
6	ERRC0	错误代码0	0	0	位错误
			0	1	格式错误
			1	0	填充错误
			1	1	其它错误
5	DIR	方向	1（RX）		接收时发生错误
			0（TX）		发送时发生错误
4	SEG4	段4	反映了当前帧片段以确定不同错误事件：		
3	SEG3	段3	00011 帧起始		
2	SEG2	段2	00010 ID28…ID21		
1	SEG1	段1	00110 ID20…ID18		
0	SEG0	段0	00100 SRTR位		
			00101 IDE位		
			00111 ID17…ID13		
			01111 ID12…ID5		
			01110 ID4…ID0		
			01100 RTR位		
			01101 保留位1		
			01001 保留位0		
			01011 数据长度代码		
			01010 数据区		
			01000 CRC序列		
			11000 CRC分隔符		
			11001 应答时间段		
			11011 应答分隔符		
			11010 帧结束		
			10010 间断		
			10001 有效错误标志		
			10110 消极代码标志		
			10011 Tolerate Dom.位		
			10111 错误分隔符		
			11100 过载标志		

总线发生错误时被迫产生相应的错误中断（中断允许时）。同时，位流处理器的当前位置被捕捉送入错误代码捕捉寄存器。其内容直到用户通过软件读出时都是不变的。读出后，捕捉机制又被激活了。访问中断寄存器期间，中断寄存器中相应的中断标志位被清除。新的总线中断直到捕捉寄存器被读出一次才可能有效。

6.4.10 错误报警界限寄存器（EMLR）

错误报警界限在该寄存器中定义。默认值（硬件复位时）为 96d。复位模式中，此寄存器对 CPU 来说是可读/写的。

表 28 错误报警界限寄存器（EWLR）的位说明；CAN 地址 13

7	6	5	4	3	2	1	0
EWL.7	EWL.6	EWL.5	EWL.4	EWL.3	EWL.2	EWL.1	EWL.0

注意，只有之前进入复位模式，EWLR 的内容才有可能被改变。直到复位模式被再次取消后，才有可能发生错误状态的改变（见状态寄存器）和由新的寄存器内容引起的错误报警中断。

12.5.15 RX 错误计数寄存器（RXERR）

RX 错误计数寄存器反映了接收错误计数器的当前值。硬件复位后寄存器被初始化为 0。工作模式中，对 CPU 来说是只读的。只有在复位模式中才可对此寄存器进行写操作。

如果发生总线关闭，RX 错误计数器就被初始化为 0。总线关闭期间，写该寄存器是无效的。

表 29 RX 错误计数器寄存器（RXERR）（CAN 地址 14）

7	6	5	4	3	2	1	0
RXERR.7	RXERR.6	RXERR.5	RXERR.4	RXERR.3	RXERR.2	RXERR.1	RXERR.0

注意，只有之前进入复位模式，才有可能由 CPU 强制 RX 错误计数器发生改变。直到复位模式被取消后，错误状态的改变（见状态寄存器）、错误报警和由新的寄存器内容引起的错误中断才可能有效。

6.4.12 TX 错误计数寄存器（TXERR）

TX 错误计数寄存器反映了发送错误计数器的当前值。工作模式中，该寄存器对 CPU 来说是只读的。只有在复位模式中才可对该寄存器进行写操作。硬件复位后，寄存器被初始化为 0。如果总线关闭，TX 错误计数器被初始化为 127 以对总线定义的最小时间（128 个总线空闲信号）计数。这段时间里读 TX 错误计数器将反映出总线关闭恢复的状态信息 。

如果总线关闭有效，向 TXERR 写入 0-254 会清除总线关闭标志，而复位模式被清除后控制器会等待一个 11 位的连续隐藏位（总线空闲）的出现。

向 TXERR 写入 255 会初始化 CPU 驱动的总线关闭事件。注意，只有之前进入复位模式，才可能由 CPU 强制改变 TX 错误计数器内容。直到复位模式被再次取消，错误或总线状态的改变（见状态寄存器）、错误报警和由新的寄存器内容引起的错误中断才有可能有效。离开复位模式后，就象总线错误引起的一样，给出新的 TX 计数器内容且总线关闭以同样的方式执行。这意味着重新进入复位模式，TX 错误计数器被初始化为 127，RX 计数器被清 0，所有的相关状态和中断寄存器位被置位。

复位模式的清除将会执行协议规定的总线关闭恢复时序（等待 128 个总线空闲信号）。

如果在总线关闭恢复（TXERR>0）之前又进入复位模式，总线关闭保持有效而 TXERR 被冻结。

表 24 TX 错误计数寄存器（TXERR）；CAN 地址 15

7	6	5	4	3	2	1	0
TXERR.7	TXERR.6	TXERR.5	TXERR.4	TXERR.3	TXERR.2	TXERR.1	TXERR.0

12.5.17 验收滤波器

在验收滤波器的帮助下，只有当接收信息中的识别位和验收滤波器预定义的值相等时，CAN 控制器才允许将已接收信息存入 RXFIFO。

验收滤波器由验收代码寄存器(ACRn)和验收屏蔽寄存器（AMRn）定义。要接收的信息的位模式在验收代码寄存器中定义。相应的验收屏蔽寄存器允许定义某些位为 “ 无关 ”(即可为任意值)。

PeliCAN 设计成支持 4 个所谓的验收滤波器分组。此外每个滤波器分组使用的帧格式可立即编程。

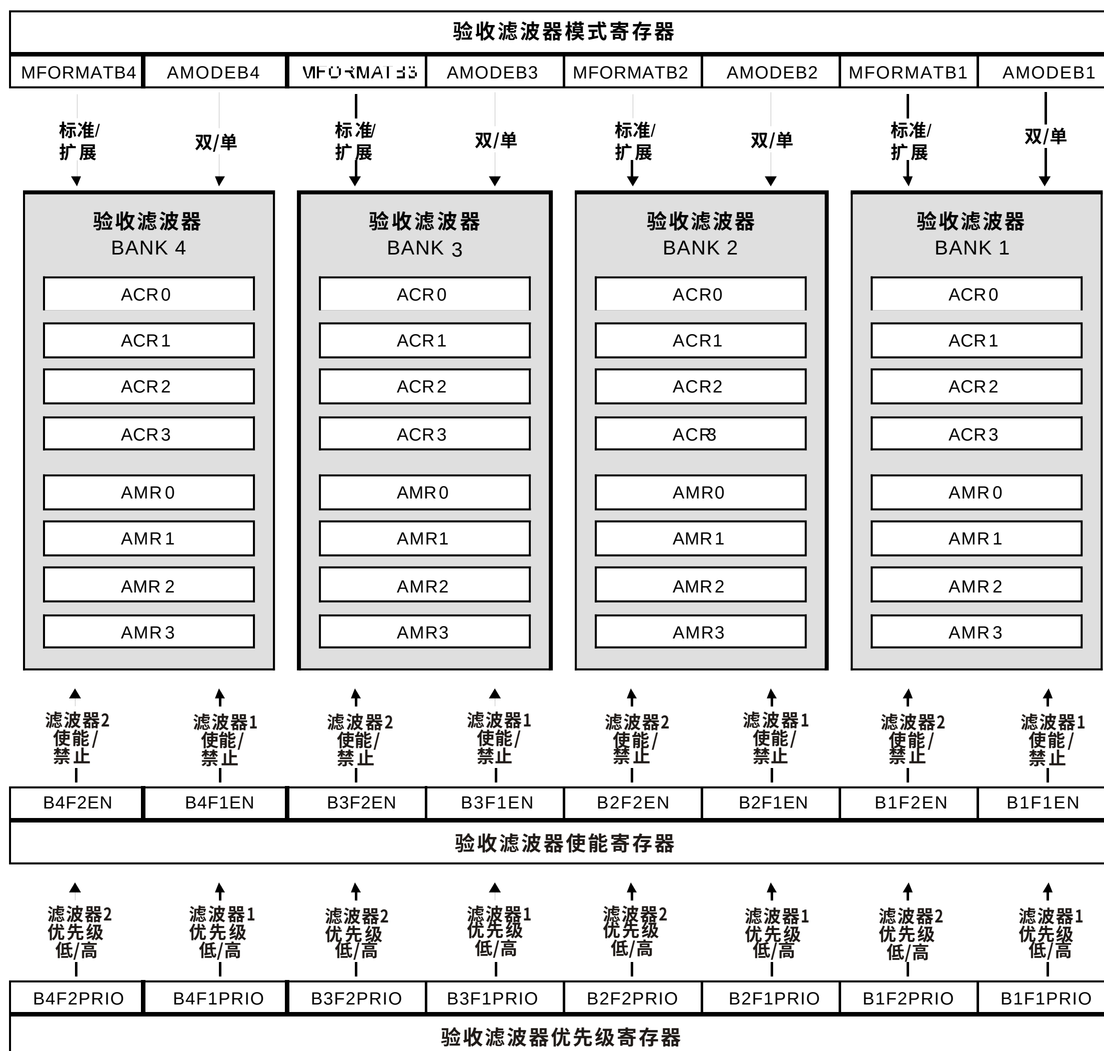


图 14 验收滤波器分组

12.5.17.1 验收滤波器模式寄存器

当前的操作模式在验收滤波器模式寄存器（CAN 地址 29）中定义。只有在复位模式中才可对该寄存器进行写操作。

表 31 验收滤波器模式寄存器（ACF 模式）（CAN 地址 29）

7	6	5	4	3	2	1	0
MFORMATB4	AMODEB4	MFORMATB3	AMODEB3	MFORMATB2	AMODEB2	MFORMATB1	AMODEB1

表 32　验收滤波器模式寄存器（ACF 模式）位说明

位	符号	名称	值	功能
7	MFORMATB4	验收滤波器格式组4	1(EFF) 0(SFF)	验收滤波器第4组仅用于扩展帧信息。标准帧信息被忽略。 验收滤波器第4组仅用于标准帧信息。扩展帧信息被忽略。
6	AMODEB4	验收滤波器模式组4	1(单) 0(双)	单验收滤波器选项使能－长滤波器有效 双验收滤波器选项使能－短滤波器有效
5	MFORMATB3	验收滤波器格式组3	1(EFF) 0(SFF)	验收滤波器第3组仅用于扩展帧信息。标准帧信息被忽略。 验收滤波器第3组仅用于标准帧信息。扩展帧信息被忽略。
4	AMODEB3	验收滤波器模式组3	1(单) 0(双)	单验收滤波器选项使能－长滤波器有效 双验收滤波器选项使能－短滤波器有效
3	MFORMATB2	验收滤波器格式组2	1(EFF) 0(SFF)	验收滤波器第2组仅用于扩展帧信息。标准帧信息被忽略。 验收滤波器第2组仅用于标准帧信息。扩展帧信息被忽略。
2	AMODEB2	验收滤波器模式组2	1(单) 0(双)	单验收滤波器选项使能－长滤波器有效 双验收滤波器选项使能－短滤波器有效
1	MFORMATB1	验收滤波器格式组1	1(EFF) 0(SFF)	验收滤波器第1组仅用于扩展帧信息。标准帧信息被忽略。 验收滤波器第1组仅用于标准帧信息。扩展帧信息被忽略。
0	AMODEB1	验收滤波器模式组1	1(单) 0(双)	单验收滤波器选项使能－长滤波器有效 双验收滤波器选项使能－短滤波器有效

12.5.17.2　验收滤波器使能寄存器

每个验收滤波器由验收滤波器使能寄存器中的相应位使能或禁止。如果之前相应的滤波器被禁止，允许在正常操作时改变验收滤波器的内容。禁止的验收滤波器不允许信息输入到接收缓冲区。如果所有的验收滤波器都被禁止（硬件复位后默认状态），将不会有信息输入到接收缓冲区。

表 33　验收滤波器使能寄存器（ACF 使能）（CAN 地址 30）

7	6	5	4	3	2	1	0
B4F2EN	B4F1EN	B3F2EN	B3F1EN	B2F2EN	B2F1EN	B1F2EN	B1F1EN

表 34 验收滤波器使能寄存器（ACF 使能）位说明

位	符号	名称	值	功能
7	B4F2EN	第4组滤波器2	1(使能)	第4组滤波器2使能，不能对相应的屏蔽和代码寄存器进行写操作
			0(禁止)	第4组滤波器2禁止，可以改变相应的屏蔽和代码寄存器
6	B4F1EN	第4组滤波器1	1(使能)	第4组滤波器1使能，不能对相应的屏蔽和代码寄存器进行写操作
			0(禁止)	第4组滤波器1禁止，可以改变相应的屏蔽和代码寄存器
5	B3F2EN	第3组滤波器2	1(使能)	第3组滤波器2使能，不能对相应的屏蔽和代码寄存器进行写操作
			0(禁止)	第3组滤波器2禁止，可以改变相应的屏蔽和代码寄存器
4	B3F1EN	第3组滤波器1	1(使能)	第3组滤波器1使能，不能对相应的屏蔽和代码寄存器进行写操作
			0(禁止)	第3组滤波器1禁止，可以改变相应的屏蔽和代码寄存器
3	B2F2EN	第2组滤波器2	1(使能)	第2组滤波器2使能，不能对相应的屏蔽和代码寄存器进行写操作
			0(禁止)	第2组滤波器2禁止，可以改变相应的屏蔽和代码寄存器
2	B2F1EN	第2组滤波器1	1(使能)	第2组滤波器1使能，不能对相应的屏蔽和代码寄存器进行写操作
			0(禁止)	第2组滤波器1禁止，可以改变相应的屏蔽和代码寄存器
1	B1F2EN	第1组滤波器2	1(使能)	第1组滤波器2使能，不能对相应的屏蔽和代码寄存器进行写操作
			0(禁止)	第1组滤波器2禁止，可以改变相应的屏蔽和代码寄存器
0	B1F1EN	第1组滤波器1	1(使能)	第1组滤波器1使能，不能对相应的屏蔽和代码寄存器进行写操作
			0(禁止)	第1组滤波器1禁止，可以改变相应的屏蔽和代码寄存器

注：如果选择单滤波器模式，该单滤波器与对应的滤波器 1 使能位相关。滤波器 2 使能位在单滤波器模式中不起作用。

12.5.17.3 验收滤波器优先级寄存器

每个可用的验收滤波器可定义为：一个信息通过特定的验收滤波器是否立即产生接收中断或可编程接收中断级是否可用于中断。这允许将特定的验收滤波器用于报警信息识别并立即向主 CPU 发出中断信号。

表 35 验收滤波器优先级寄存器（ACF 优先级）（CAN 地址 31）

7	6	5	4	3	2	1	0
B4F2PRIO	B4FIPRIO	B3F2PRIO	B3F1PRIO	B2F2PRIO	B2F1PRIO	B1F2PRIO	B1F1PRIO

表 36 验收滤波器优先级寄存器（ACF 优先级）位说明

位	符号	名称	值	功能
7	B4F2PRIO	第4组滤波器2优先级	1(高) 0(低)	如果信息通过第4组滤波器2，立即产生接收中断 如果FIFO级超过接收中断级滤波器，产生接收中断
6	B4FIPRIO	第4组滤波器1优先级	1(高) 0(低)	如果信息通过第4组滤波器1，立即产生接收中断 如果FIFO级超过接收中断级滤波器，产生接收中断
5	B3F2PRIO	第3组滤波器2优先级	1(高) 0(低)	如果信息通过第3组滤波器2，立即产生接收中断 如果FIFO级超过接收中断级滤波器，产生接收中断
4	B3F1PRIO	第3组滤波器1优先级	1(高) 0(低)	如果信息通过第3组滤波器1，立即产生接收中断 如果FIFO级超过接收中断级滤波器，产生接收中断
3	B2F2PRIO	第2组滤波器2优先级	1(高) 0(低)	如果信息通过第2组滤波器2，立即产生接收中断 如果FIFO级超过接收中断级滤波器，产生接收中断
2	B2F1PRIO	第2组滤波器1优先级	1(高) 0(低)	如果信息通过第2组滤波器1，立即产生接收中断 如果FIFO级超过接收中断级滤波器，产生接收中断
1	B1F2PRIO	第1组滤波器2优先级	1(高) 0(低)	如果信息通过第1组滤波器2，立即产生接收中断 如果FIFO级超过接收中断级滤波器，产生接收中断
0	B1F1PRIO	第1组滤波器1优先级	1(高) 0(低)	如果信息通过第1组滤波器1，立即产生接收中断 如果FIFO级超过接收中断级滤波器，产生接收中断

12.5.17.4 单滤波器配置

在该滤波器配置中可以定义一个长滤波器（4 字节）。滤波器字节和信息字节之间位的对应关系取决于已编程的帧格式（见 ACF 模式寄存器）。

单滤波器标准帧：如果选择的是标准帧格式，使用包括 RTR 位的完整的识别码和头两个数据字节进行验收滤波。如果由于置位 RTR 位而导致没有数据字节，或因为设置相应的数据长度代码而没有或只有一个数据字节，信息也会被接收到。

对于一个成功接收的信息，所有单个位的比较后都必须发出接受信号。注意，AMR1 和 ACR1 的低四位是不用的。为了和将来的产品兼容，这些位可通过置位 AMR1.3、AMR1.2、AMR1.1 和 AMR1.0 定义为“无关”。

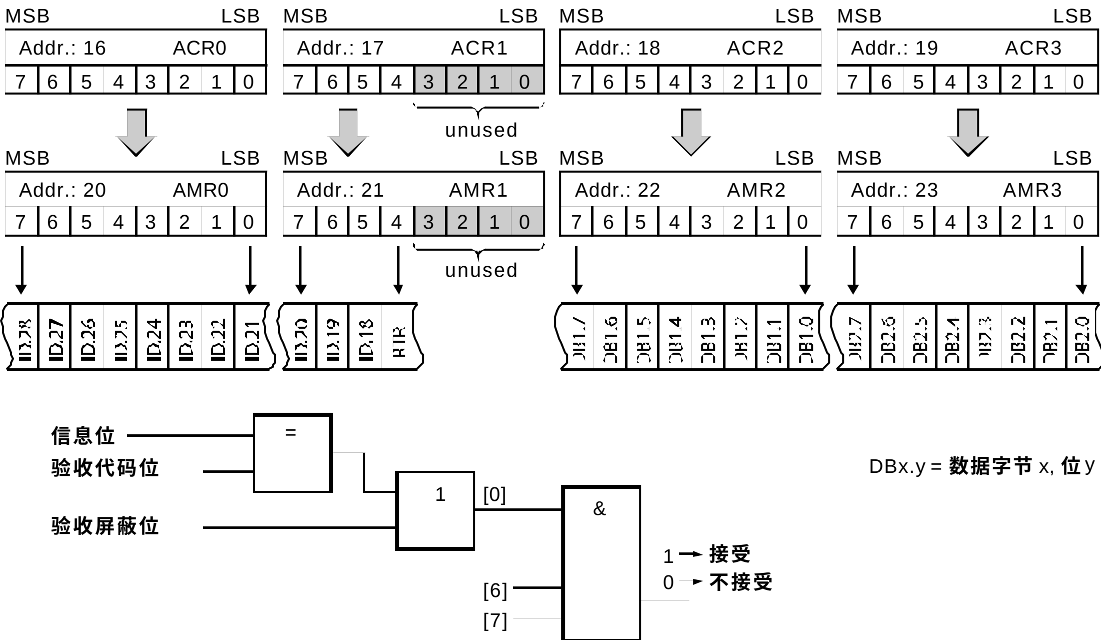


图 15 接收标准帧信息时的单滤波器配置

单滤波器扩展帧：如果选择扩展帧格式，包括 RTR 位的全部识别码将用于验收过滤。

为了成功接收信息，每个位的比较后都必须发出接受信号。必须注意的是，AMR3 的最低两位和 ACR3 是不用的。为了和将来的产品兼容，这些位应该通过置位 AMR3.1 和 AMR3.0 定义为“无关”。

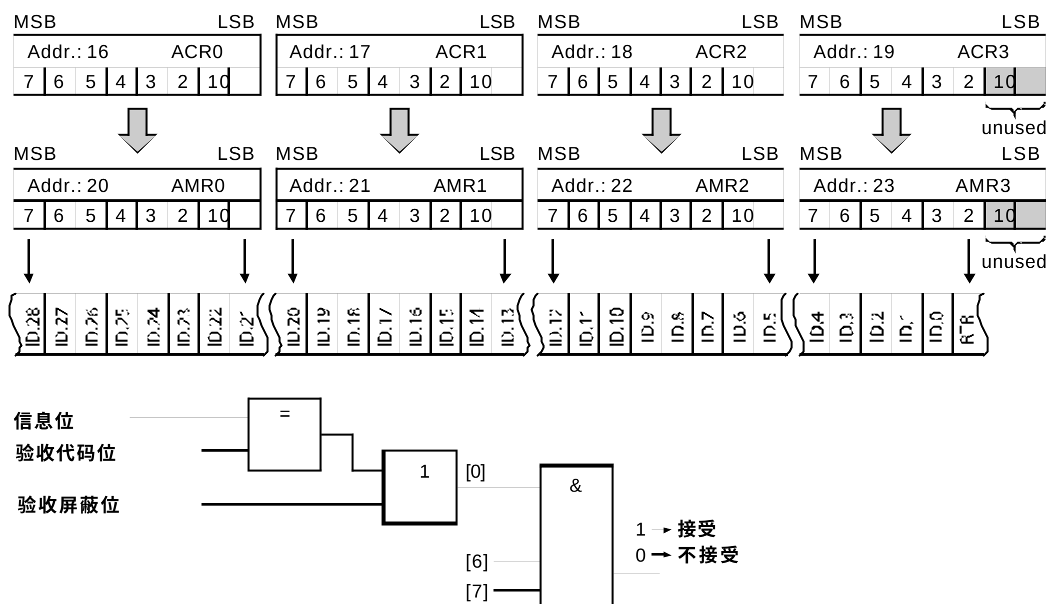


图 16　接收扩展帧信息的单滤波器配置

12.5.17.5　双滤波器的配置

该滤波器配置可以定义两个短滤波器。一条接收的信息要和两个滤波器比较来决定是否放入接收缓冲器中。至少有一个滤波器发出接受信号，接收的信息才有效。滤波器字节和信息字节之间位的对应关系取决于当前接收的帧格式。

双滤波器标准帧：如果选择标准帧格式，被定义的两个滤波器是不一样的。第一个滤波器比较包括 RTR 位的整个标准识别码和信息的第一个数据字节。第二个滤波器只比较包括 RTR 位的整个标准识别码。

为了成功接收信息，所有单个位的比较时应至少有一个滤波器表示接受。RTR 位置位或数据长度代码为 0 时表示没有数据字节存在。无论怎样，只要从开始到 RTR 位的部分都被表示接收，信息就可以通过滤波器 1。

如果滤波器 1 不需要过滤数据字节，AMR1 和 AMR3 的低四位必须被置为 1（无关）。这样两个滤波器使用包括 RTR 位的整个标准识别码进行同样的工作。

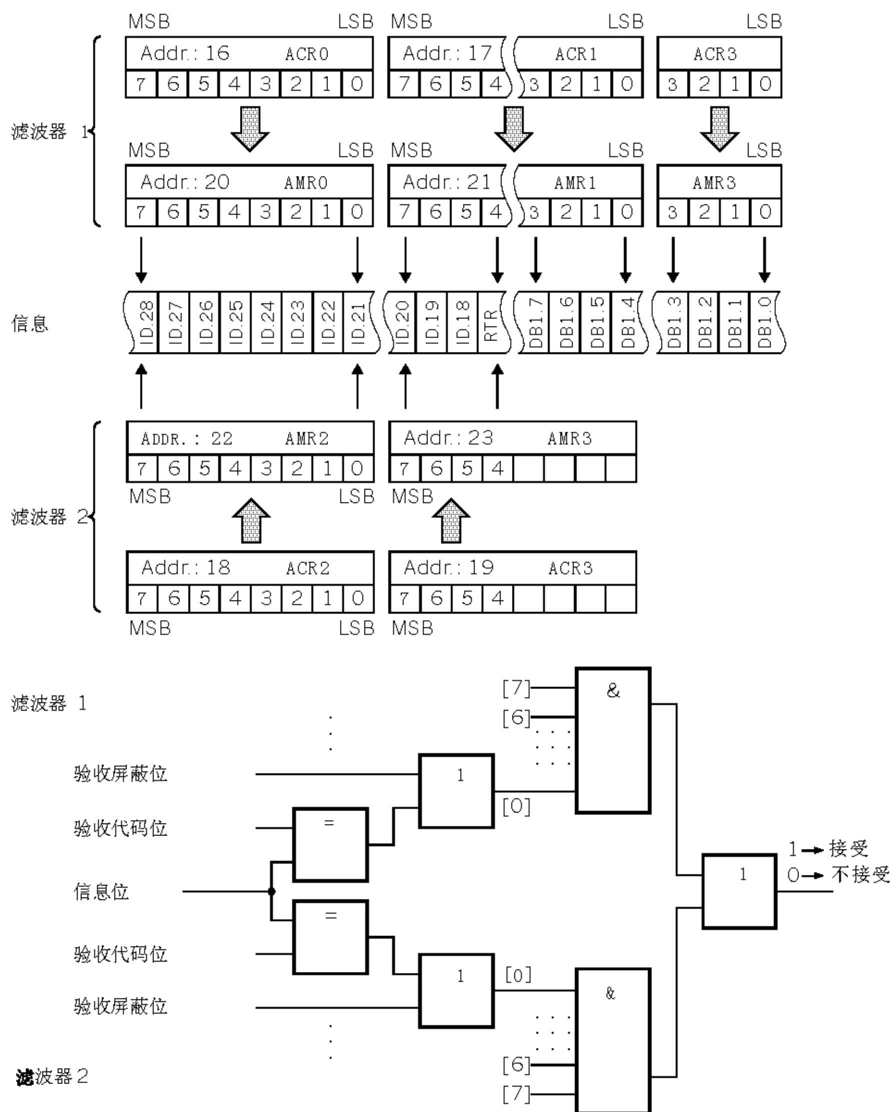


图 17　接收标准帧信息的双滤波器配置

双滤波器扩展帧：如果选择扩展帧格式，两个定义的滤波器看起来是相同的。两个滤波器都只比较扩展识别码的前两个字节。

为了能成功接收信息，所有单个位的比较时至少有一个滤波器表示接收。

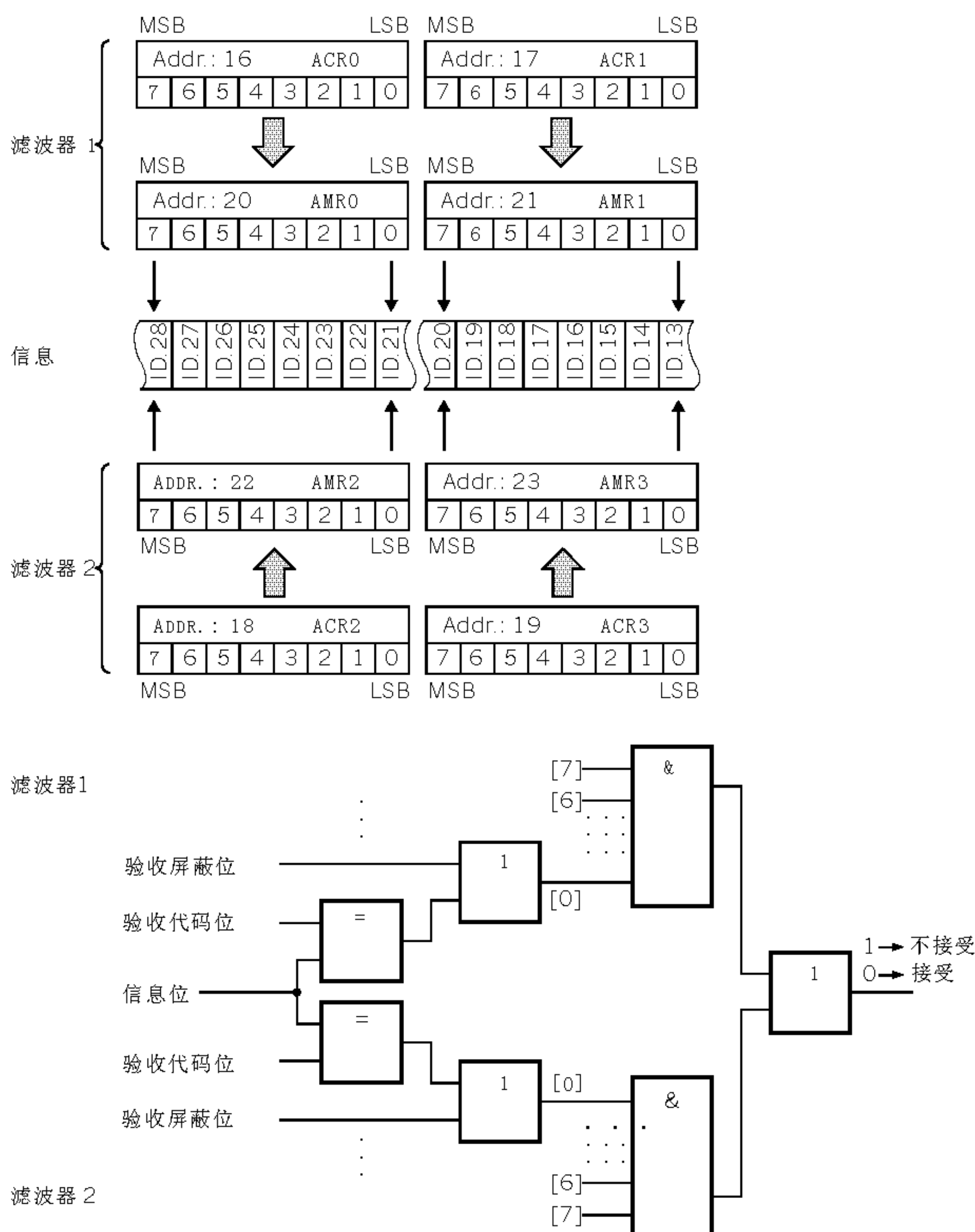


图 18　双滤波器配置，接收扩展帧信息

6.4.13 发送缓冲器

发送缓冲器的整体布局见图 19。请务必区分标准帧格式（SFF）和扩展帧格式（EFF）配置。发送缓冲器允许定义最长为 8 个数据字节的发送信息。

6.4.13.1 **发送缓冲器布局**

发送缓冲器分为描述符区和数据区，描述符区的第一个字节是帧信息字节（帧信息）。它说明了帧格式（SFF 或 EFF）、远程或数据帧和数据长度。SFF 有两个字节的识别码，EFF 有四个字节的识别码。数据区最多容纳 8 个数据字节。发送缓冲器长 13 个字节，在 CAN 地址的 112~124。

标准帧格式（SFF）

CAN 地址	
112	TX 帧信息
113	TX 识别码 1
114	TX 识别码 2
115	TX 数据字节1
116	TX 数据字节 2
117	TX 数据字节 3
118	TX 数据字节 4
119	TX 数据字节 5
120	TX 数据字节 6
121	TX 数据字节 7
122	TX 数据字节 8
123	unused
124	unused

扩展帧格式（EFF）

CAN 地址	
112	TX 帧信息
113	TX 识别码 1
114	TX 识别码 2
115	TX 数据字节 3
116	TX 数据字节 4
117	TX 数据字节 1
118	TX 数据字节 2
119	TX 数据字节 3
120	TX 数据字节 4
121	TX 数据字节 5
122	TX 数据字节 6
123	TX 数据字节 7
124	TX 数据字节 8

图 19 标准帧和扩展帧格式配置的发送缓冲器配置

6.4.13.2 发送缓冲器的描述符区

发送缓冲器位的分布见图 20。选择的配置和接收缓冲器分布兼容。

标准帧格式(SFF)

Addr. 112 TX 帧信息

7	6	5	4	3	2	1	0
FF	RTR	(0)	(0)	DLC.3	DLC.2	DLC.1	DLC.0

Addr. 113 TX 识别码 1

7	6	5	4	3	2	1	0
ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21

Addr. 114 TX 识别码 2

7	6	5	4	3	2	1	0
ID.20	ID.19	ID.18	(RTR)	(0)	(0)	(0)	(0)

发送缓冲区位的含义:

ID.x	识别码位x
FF	帧格式
RTR	远程发送请求
DLC.x	数据长度代码位 x
X	无关
(0)	无关，但推荐与接收缓冲区兼容

扩展帧格式(EFF)

Addr. 112 TX 帧信息

7	6	5	4	3	2	1	0
FF	RTR	(0)	(0)	DLC.3	DLC.2	DLC.1	DLC.0

Addr. 113 TX 识别码 1

7	6	5	4	3	2	1	0
ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21

Addr. 114 TX 识别码 2

7	6	5	4	3	2	1	0
ID.20	ID.19	ID.18	ID.17	ID.16	ID.15	ID.14	ID.13

Addr. 115 TX 识别码 3

7	6	5	4	3	2	1	0
ID.12	ID.11	ID.10	ID.9	ID.8	ID.7	ID.6	ID.5

Addr. 116 TX 识别码 4

7	6	5	4	3	2	1	0
ID.4	ID.3	ID.2	ID.1	ID.0	(RTR)	(0)	(0)

图 20 发送缓冲区的位分布

表 37 帧格式（FF）和远程发送请求（RTR）位

位	值	功能
FF	1 (EFF)	通过CAN控制器发送扩展帧格式
	0 (SFF)	通过CAN控制器发送标准帧格式
RTR	1(远程)	通过CAN控制器发送远程帧
	0(数据)	通过CAN控制器发送数据帧

12.5.18.3 数据长度代码（DLC）

数据区的信息字节长度由数据长度代码编制。在远程帧发送开始时由于 RTR 位被置位（远程），数据长度代码是不被考虑的。这使接收/发送的数据字节数目为 0。如果有两个 CAN 控制器使用同一个识别码同时启动远程帧传送，数据长度代码必须正确说明以避免总线错误。

数据字节长度范围是 0-8，编码形式如下：

数据字节数=8×DLC.3+4×DLC.2+2×DLC.1+DLC.0

为了兼容，大于 8 的数据长度代码是不可用的。如果大于 8，将以 8 个字节计。

12.5.18.4 识别码（ID）

标准帧格式（SFF）的识别码有 11 位（ID.28-ID.18），扩展帧格式的识别码有 29 位（ID.28-ID.0）。ID.28 是最高位，在总线仲裁过程中最先发送到总线上。识别码就象信息的名字一样，使用在验收滤波器中，而且在仲裁过程中决定了总线访问的优先权。识别码的二进值越低优先权越高。这是由于仲裁时有更大数字的前导支配位。

12.5.18.5 数据区

发送的字节数取决于数据长度代码。最先发送的是在 CAN 地址 115（SFF）或 117（EFF）的数据字节 1 的最高位。

12.5.19 接收缓冲器

接收缓冲器的整个布局与前面一节讲述的发送缓冲器很相似。接收缓冲器是 RXFIFO 的可访问部分，位于 CAN 地址的 96 和 108 之间。每条信息都分为描述符和数据区。

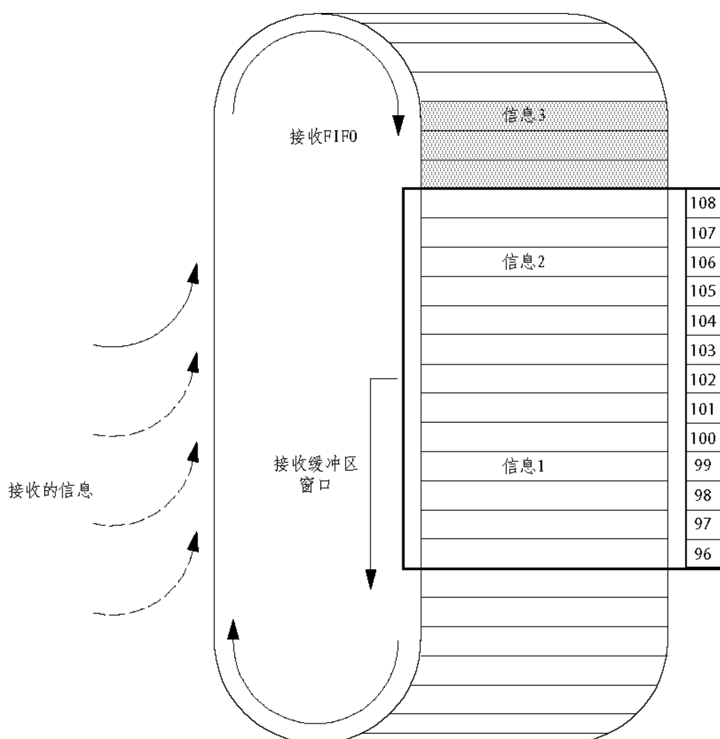
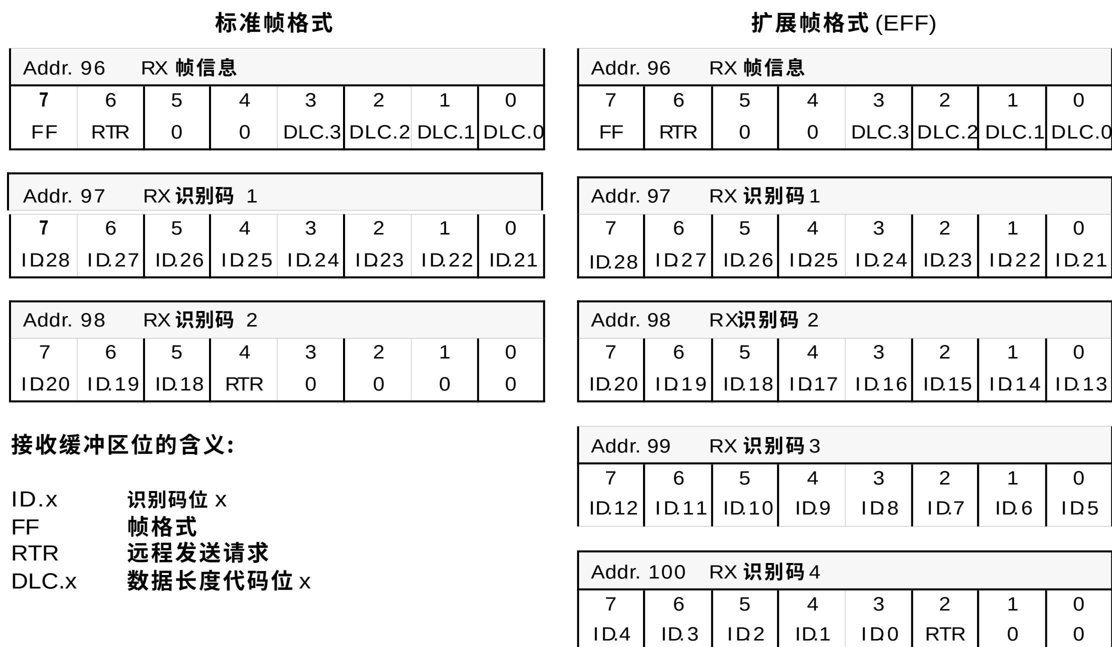


图 21　RXFIFO 中的信息存储举例

12.5.19.1 接收缓冲器的描述符文件

识别码、帧格式、远程发送请求位和数据长度代码和在发送缓冲器中描述的含义相同。



标准帧格式

Addr. 96 RX 帧信息							
7	6	5	4	3	2	1	0
FF	RTR	0	0	DLC.3	DLC.2	DLC.1	DLC.0

Addr. 97 RX 识别码 1							
7	6	5	4	3	2	1	0
ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21

Addr. 98 RX 识别码 2							
7	6	5	4	3	2	1	0
ID.20	ID.19	ID.18	RTR	0	0	0	0

接收缓冲区位的含义:

ID.x	识别码位 x
FF	帧格式
RTR	远程发送请求
DLC.x	数据长度代码位 x

扩展帧格式 (EFF)

Addr. 96 RX 帧信息							
7	6	5	4	3	2	1	0
FF	RTR	0	0	DLC.3	DLC.2	DLC.1	DLC.0

Addr. 97 RX 识别码 1							
7	6	5	4	3	2	1	0
ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21

Addr. 98 RX识别码 2							
7	6	5	4	3	2	1	0
ID.20	ID.19	ID.18	ID.17	ID.16	ID.15	ID.14	ID.13

Addr. 99 RX 识别码 3							
7	6	5	4	3	2	1	0
ID.12	ID.11	ID.10	ID.9	ID.8	ID.7	ID.6	ID.5

Addr. 100 RX 识别码 4							
7	6	5	4	3	2	1	0
ID.4	ID.3	ID.2	ID.1	ID.0	RTR	0	0

图 22 接收缓冲器的位分布

注：在帧信息字节中的接收字节长度代码代表实际发送的数据长度代码，它有可能大于 8（取决于发送器）。无论如何，最大接收数据字节数是 8。这一点在读接收缓冲器中的信息时应当考虑。

RXFIFO 中一次可以存储多少 CAN 信息取决于数据的长度。如果 RXFIFO 中没有足够的空间来存储新的信息，CAN 控制器会产生数据溢出条件，此时信息有效且验收检测为肯定。发生数据溢出情况时，已部分写入 RXFIFO 的信息将被删除。这种情况可以通过状态寄存器和数据超限中断（如果中断使能）通知 CPU。

13 串行 I/O

PP8xC591 带有 3 个独立的串行接口：CAN、SIO0 和 SIO1。SIO0 是一个具有增强功能的标准 UART 串口。在下面将会有一章讲述标准 UART 功能而由额外的一章讲述增强型 UART。SIO1 为 I^2C 总线。

14 SIO0 标准串行接口 UART

串口为全双工结构，表示可以同时发送和接收。它还具有接收缓冲，意味着在第一个字节从寄存器读出之前，可以开始接收第二个字节。(但是如果第二个字节接收完毕时第一个字节仍未读出，其中一个字节将会丢失）。串口的发送和接收寄存器都是通过 SFR S0BUF 进行访问的。写入 S0BUF 的数据装入发送寄存器，对 S0BUF 的读操作是对物理上分开的接收寄存器进行访问。

串口有 4 种操作模式（一种同步模式，三种异步模式)。串口波特率时钟是由振荡频率决定的（模式 0 和 2)，或者由定时器 1 或专门的波特率发生器产生（模式 1 和 3)：

模式 0 移位寄存器模式（同步）

串行数据通过 RxD 进出。TxD 输出时钟。每次发送或接收以 LSB（最低位）作首位，每次 8 位。波特率固定为 MCU 时钟频率的 1/6。

模式 1 8 位 UART 可变波特率

TxD 脚发送，RxD 脚接收，每次数据为 10 位，一个起始位（0)，8 个数据位（LSB 在前）及一个停止位（1)。当接收数据时，停止位存于 SCON 的 RB8 内，波特率可变。

模式 2 9 位 UART 固定波特率

TxD 脚发送，RxD 脚接收，每次数据为 11 位，一个起始位（0），8 个数据位（LSB 在前），一个可编程第 9 位数据及一个停止位（1）。

发送时，第 9 个数据位（SCON 内 TB8 位）可置为 0 或 1。例如将奇偶位（PSW 内 P 位）移至 TB8。接收时，第 9 位数据存入 SCON 的 RB8 位，停止位忽略。波特率可编程为 MCU 时钟频率的 1/16 或 1/32，由 PCON 内 SMOD1 位决定。

模式 3 9 位 UART 可变波特率

TxD 脚发送，RxD 脚接收，每次数据为 11 位，一个起始位（0），8 个数据位（LSB 为首位），一可编程的第 9 位数据及一个停止位（1）。事实上模式 3 除了波特率外均与模式 2 相同。其波特率可变。

在上述 4 种模式中，发送过程是以任意一条以写 SBUF 作为目标寄存器的指令开始的，模式 0 时接收通过设置 R1=0 及 REN=1 初始化，其它模式下如若 REN=1 则通过起始位初始化。

多机通信

UART 模式 2 及模式 3 有一个专门的应用领域即多机通信。在这些模式时，接收为 9 位数据。第 9 位存入 RB8。接下来为停止位。UART 可编程为：接收到停止位时，仅当 RB8=1 时串口中断才有效。可通过置位 SCON 内 SM2 位来选择这一特性。下述为多机系统利用这一特性的一种方法。

当主机需要发送一数据块给数台从机之一时，首先发送出一个地址字节对目标从机进行识别。地址与数据字节通过第 9 位数据区别，其中地址字节的第 9 位为 1，而数据字节为 0。SM2=1 时，数据字节不会使各从机产生中断，而地址字节则令所有从机中断，这样各从机可以检查接收到的数据判断是否被寻址。被寻址的从机即可清除 SM2 位以准备接收随后数据内容。未被寻址的从机的 SM2 位仍为 1 则不理睬随后数据继续各自工作。

模式 0 时 SM2 无效，模式 1 时 SM2 用于检验停止位是否有效。在模式 1 时，如果 SM2=1，那么只有接收到有效的结束位才可产生接收中断。

串口控制寄存器

串行端口控制及状态寄存器即 SCON，如图 12 所示，其中包括模式选择位，以及发送和接收的第 9 位数据（TB8 及 RB8），以及串口中断位（TI 及 RI）。S0BUF 是串口的接收和发送缓冲器，写入 S0BUF 的数据装入发送寄存器并初始化发送，对 S0BUF 的读操作是对物理上分开的接收寄存器进行访问。

波特率

用于串口的波特率时钟有几种方式产生，它取决于所处的操作模式。有必要说明一下“波特率时钟”和“波特率”的不同。串口要求时钟频率为波特率的 16 倍用于内部的同步。因此，波特率发生器必须向串口提供一个“波特率时钟”，将其 16 分频之后就得到实际的“波特率”。但是下面章节所给出的所有公式都已包含了这一项并计算最后的波特率。此外，缩写 f_{CLK} 是指外部时钟频率（振荡器或外部输入时钟）

串口的波特率是由两个位 SPS 和 SMOD1 决定的，它们分别位于寄存器 S0PSH 和 PCON 中。在 S0PSH 和 S0PSL 中，内部波特率发生器的预分频装入值是可以编程的（见表 38~43）。

内部波特率发生器预分频器 S0PSH、S0PSL

表 38 内部波特率发生器预分频器低字节 S0PSL（地址 FAH）

7	6	5	4	3	2	1	0
预分频器装入值							

表 39 S0PSL 位描述

位	符号	描述
7~0	—	波特率重装值低字节：波特率定时器重装值低 8 位

表 40 内部波特率发生器预分频器高字节 S0PSH（地址 FBH）

7	6	5	4	3	2	1	0
SPS	–	–	–	预分频器装入值			

表 41 S0PSH 位描述

位	符号	描述
7	SPS	波特率发生器使能：置位时由专门的波特率发生器产生波特率，清零时，波特率由定时器 1 溢出速率决定。
6~4	–	保留
3~0	–	波特率重装值：波特率定时器重装值高 4 位

表 42 PCON（地址 87H）

7	6	5	4	3	2	1	0
SMOD1	SMOD0	(POF)	(WLE)	(GF1)	(GF0)	(PD)	(IDL)

表 43 SMOD1 和 SMOD0 位描述

位	符号	描述
7	SMOD1	双倍波特率 当置位时,串口模式 1,2 和 3 的波特率加倍,复位后该位清零
6	SMOD0	选择 SM0/FE(SCON.7)
5~0	(POF)~(IDL)	细节参见电源控制寄存器(PCON)章节

波特率产生选项概述

根据所编程的操作模式，可通过几种不同的途径产生波特率时钟。图 23 所示为串口波特率时钟产生的两个控制位和由特殊功能寄存器 SCON 选择的模式。

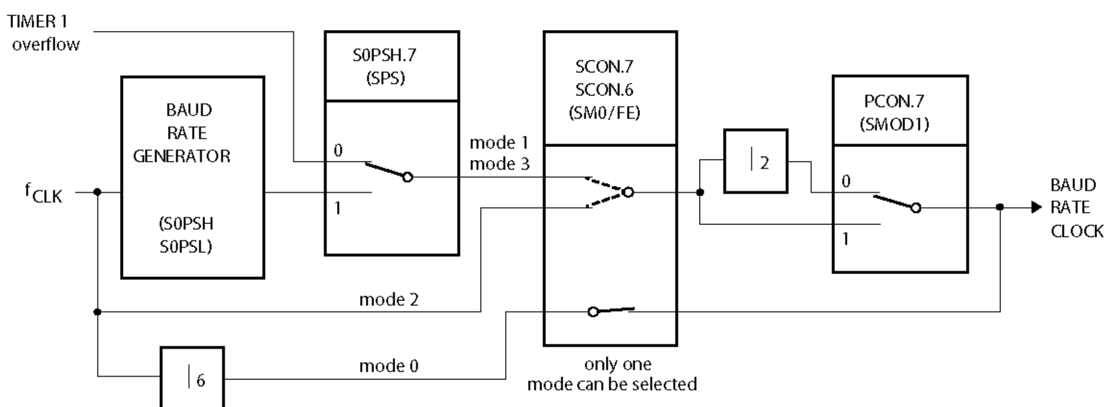


图 23 串口波特率的产生

14.3.4 模式 0 中的波特率

操作模式 0 的波特率是固定的。为 fosc/6。

14.3.5 模式 2 中的波特率

模式 2 中的波特率是 MCU 时钟/16 或 MCU 时钟/32，取决于 PCON 寄存器中的 SMOD1 位的值。若 SMOD1=0（复位后的值），波特率为振荡器频率的 1/32，若 SMOD1=1，波特率为振荡器频率的 1/16。

14.3.6 模式 1 和模式 3 中的波特率

模式 1 和模式 3 的波特率可变并选择由波特率发生器或定时器 1 产生。

14.3.7 使用内部波特率发生器

在模式 1 和模式 3 中，PP8xC591 可使用内部波特率发生器。要使能该功能，位 SPS（S0PSH.7）必须置位。位 SMOD1（PCON.7）控制影响波特率发生器输入和输出时钟的一个 2 分频电路。复位后该 2 分频电路有效，这使得溢出输出时钟被 2 分频。波特率发生器的输入时钟为 fclk。

波特率发生器包括它自身独立运行的递增计数的 12 位定时器。该定时器溢出时（计数值 FFFH 之后的下个计数），从寄存器 S0PSL 和 S0PSH 自动重装 12 位值。定时器的低 8 位取自 S0PSL。而高 4 位取自寄存器 S0PSH 的位 0~3。波特率定时器的值通过对 S0PSH 的写操作重新装入。

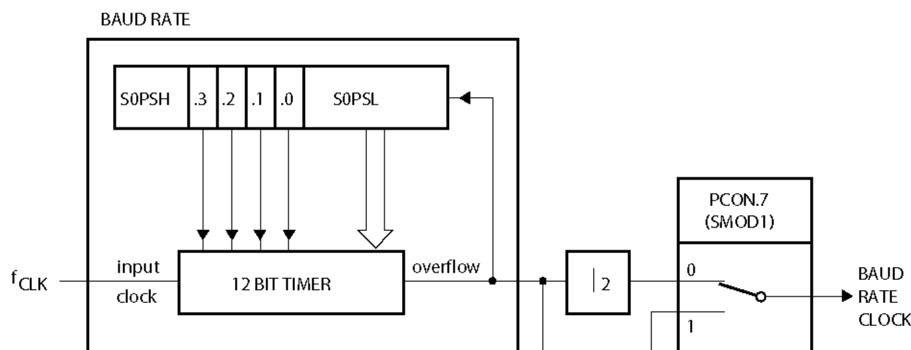


图 24 使用波特率发生器时的串口输入时钟

当波特率发生器作为串口模式 1 和模式 3 的时钟源时，波特率由下式确定：

$$\text{模式1, 3波特率} = \frac{2^{SMOD1} \times \text{振荡器频率}}{32 \times \text{波特率溢出速率}}$$

S0PS：波特率发生器预分频器装载值。

表 47 列出了通过内部波特率发生器产生的波特率以及它们是如何获得的。

14.3.8 使用定时器 1 作波特率发生器

在串口模式 1 和 3 中，定时器 1 可用于产生波特率。波特率由定时器 1 的溢出速率和 SMOD1 的值决定。

$$\text{模式1, 3波特率} = \frac{2^{SMOD1}}{32} \times (\text{定时器1溢出速率})$$

在此应用中定时器 1 中断通常被禁止，定时器 1 可以工作在定时或计数方式和 3 种工作模式中任何一个。在最典型应用中，它用作定时器方式工作自动重装载模式（TMOD 的高半字节为 0010B），它的波特率值由下式给出：

$$\text{模式1, 3波特率} = \frac{2^{SMOD1} \times \text{振荡器频率}}{32 \times 6 \times (256-(TH1))}$$

可以定时器 1 的中断实现非常低的波特率。将定时器配置为 16 位定时器（TMOD 的高半字节为 0001B），并使用中断进行 16 位软件重装。表 49 列出了几个常用的波特率以及如何从定时器 1 获得。

表 44 串口控制寄存器 S0CON(地址 98H)

7	6	5	4	3	2	1	0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

表 45 S0CON 位描述

位	符号	描述
7	SM0	串口模式位 0，见表 46
6	SM1	用于选择串口模式，见表 46
5	SM2	使能模式 2 和 3 的多机通信特性　在模式 2 和 3 中，如果 SM2=1，且接收到的第 9 位数据（RB8）是 0，则 RI（接收中断标志）不会被激活。在模式 1 中，如果 SM2=1 且没有接收到有效的停止位，则 RI 不会被激活。在模式 0 中，SM2 必须是 0。
4	REN	使能串口接收。由软件置位或清除。REN=1 时，使能接收，REN=0 时，禁止接收。
3	TB8	模式 2 和 3 中发送的第 9 位数据，可以按需要由软件置位或清除。
2	RB8	模式 2 和 3 中已接收的第 9 位数据，在模式 1 中，或 SM2=0，RB8 是已接收的停止位。在模式 0 中，RB8 未用。
1	TI	发送中断标志。模式 0 中。在发送完第 8 位数据时，由硬件置位。其它模式中，在发送停止位之初，由硬件置位。在任何模式中，都必须由软件来清除 TI。
0	RI	接收中断标志，模式 0 中，接收第 8 位结束时由硬件置位。其它模式中，在接收停止位的中间时刻，由硬件置位。在任何模式(SM2 所述情况除外)必须由软件清除 RI。

表 46 串口模式选择

SM0	SM1	模式	描述	波特率
0	0	模式 0	移位寄存器	1/6×f_{CLK}
0	1	模式 1	8 位 UART	可变
1	0	模式 2	9 位 UART	1/32 或 1/16×f_{CLK}
1	1	模式 3	9 位 UART	可变

表 47 内部波特率定时器所产生的波特率

波特率 (KBits/s)	fCLK(MHz)	SPS	SMOD1	内部波特率定时器		
				偏差%	模式	重装值
750	12	1	1	0	1/3	FFFh
500	8	1	1	0	1/3	FFFh
250	8	1	0	0	1/3	FFFh
250	8	1	1	0	1/3	FFEh
57.6	12	1	1	0.16	1/3	FF3h
38.4	8	1	1	0.16	1/3	FF3h
19.2	12	1	1	0.16	1/3	FD9h
9.6	12	1	1	0.16	1/3	FB2h
4.8	12	1	1	0.16	1/3	F64h
2.4	12	1	1	0.16	1/3	EC8h
0.11	8	1	0	-0.01	1/3	71Fh

表 48 定时器 1 产生的波特率

波特率 (KBits/s)	fCLK(MHz)	SPS	SMOD1	内部波特率定时器		
				偏差%	模式	重装值
110	12	0	0	0.03	1	FDC8h
110	4	0	1	-0.06	1	FE85h
110	4	0	0	0.21	2	43h

14.4 有关 UART 模式的更多内容

UART 模式 0

串行数据由 RxD 端出入。TxD 输出同步移位时钟，发送或接收的是 8 位数据，低位在先，其波特率固定为 MCU 时钟的 1/6，图 25 为串行口模式 0 的简略功能框图及相关的时序。

执行任何一条把 SBUF 作为目的寄存器的指令时，就开始发送。在 S6P2 时刻的“写 SBUF”信号将 1 装入发送移位寄存器的第 9 位，并通知 TX 控制模块开始发送。写 SBUF 信号有效的一个完整的机器周期后 SEND 端有效。

SEND 使能 RxD（P3.0）端送出数据，TxD（P3.1）输出移位时钟。每个机器周期的 S3、S4 及 S5 状态时移位时钟为低电平，而 S6、S1 及 S2 时为高电平。在 SEND 有效时，每一机器周期的 S6P2 时刻发送移位寄存器的内容右移一位。

数据位向右移时，左边添加零。当数据字节最高位（MSB）移到移位寄存器的输出端时，其左边是装入“1”的第 9 位，再左的内容均为 0，此时通知 Tx 控制模块进行最后一位移位处理后禁止 SEND 并置位 T1，所有这些步骤均在“写入 SBUF”后第 10 个机器周期的 S1P1 时进行的。接收初始化条件是 REN=1 及 R1=0。下一机器周期的 S6P2 时，RX 控制单元向接收移位寄存器写入 1111 1110 并在下一个时钟使 RECEIVE 端有效。

RECEIVE 使能 P3.1 输出移位时钟，移位时钟在每个机器周期的 S3P1 及 S6P1 发生跳变。在 RECEIVE 有效时每一机器周期的 S6P2 时刻，接收移位寄存器内容向左移一位。从右移位进来的值是该机器周期 S5P2 时从 P3.0 脚上采样得来的。

数据从右边移入时，左边移出为“1”。当初始时置入最右端的“0”移至最左端时，通知 RX 控制模块作最后一次移位后装入 SBUF。在写入 SCON 清除 R1 后第 10 个机器周期的 S1P1，由于 RI 置位，RECEIVE 端被清除。

UART 模式 1

串行口工作于模式 1 时，传输的是 10 位：1 位起始位（0），8 位数据（低位在先）及一位停止位（1）。由 RxD 接收，TxD 发送。接收时，停止位存入 SCON 内 RB8。80C51 波特率取决于定时器 1 的溢出速率。图 26 所示为串行口模式 1 的功能简图及相应的发送/接收时序。

发送过程是由执行任意一条以 SBUF 为目的寄存器的指令启动的。“写 SBUF”信号还把 1（TB8）装入发送移位寄存器的第 9 位，同时通知 TX 控制单元进行发送。实际上发送过程开始于 16 分频计数器下次翻转后的那个机器周期的 S1P1 时刻。每位的发送时序与 16 分频计数器同步，而并不与“写 SBUF”信号同步。

发送以激活 SEND 端开始，向 TxD 发送一起始位。一个位（时间）以后 DATA 端有效，使输出移位寄存器中数据得以送至 TxD。再过一个位时间，产生第一个移位脉冲。

数据向右移出，左边不断填以 0，当数据字节的最高位移到移位寄存器的输出位置时，其左边是装入“1”的第 9 位，再左的内容均为 0。此时通知 TX 控制器作最后一次移位，然后禁止 SEND 端并置位 TI。这都发生于“写 SBUF”后 16 分频计时器的第 10 次翻转时。

接收在 RxD 端检测到负跳变时启动，为此 MCU 对 RxD 不断采样，采速率为波特率的 16 倍。当检测到负跳变时，16 分频计数器立即复位，同时将 1FFH 写入输入移位寄存器。复位 16 分频计时器确保计时器翻转时位与输入数据位时间同步。

计数器的 16 个状态将每个位时间分为 16 份。在第 7、8、9 计数器状态时，位检测器对 RxD 端的值采样。取值为三个采样值中取多数（至少 2 个）作为读入值，这样可以抑制噪声。如果所接收的第一位不为 0，说明它不是一帧数据的起始位，该位被摒弃，接收电路被复位，等待另一个负跳变的到来。这用来防止错误的起始位。如果起始位有效，则被移入输入移位寄存器，并开始接收这一帧中的其它位。

当数据位逐一由右边移入时，“1”从左边被移出。当起始位 0 移到最左边时（模式 1 为 9 位寄存器），通知接收控制器进行最后一次移位，将移位寄存器内容（9）位分别装入 SBUF 及 RB8，并置 RI=1。仅当最后一位移位脉冲产生时同时满足下述 2 个条件：① RI=0，②SM2=0 或接收到的停止位=1，才会装载 SBUF

和 RB8，并且置位 RI。

上述两个条件任一不满足,所接收到的数据帧就会丢失，不再恢复。两者都满足时，停止位就进入 RB8，8 位数据进入 SBUF，RI=1。这时，无论上述条件满足与否，接收控制单元都会重新等待 RxD 的负跳变。

模式 2 和模式 3

模式 2 和 3 中，发送（通过 TxD）和接收（通过 RxD）都是 11 位，包括 1 位起始位（0），8 位数据位（LSB 在先），1 位可编程数据位（第 9 位）及一位停止位（1）。发送时，第 9 位数据位（TB8）可置为 0 或 1。接收时，第 9 位存入 SCON 的 RB8。模式 2 时波特率可编程选为 MCU 时钟频率的 1/16 或 1/32。模式 3 时可由定时器 1 获取可变的波特率。

图 27 所示为模式 2、3 时串行口的功能简图。接收部分与模式 1 相同。发送部分仅发送移位寄存器内第 9 位和模式 1 有所不同。

发送过程是由执行一条以 SBUF 为目的寄存器的指令启动的。“写 SBUF”同时将 TB8 装入发送移位寄存器的第 9 位位置上。并通知发送控制器进行一次发送。发送过程由于 16 分频计数器下一次翻转后机器周期的 S1P1 时刻开始。

发送过程由使能 SEND 有效开始，将一个起始位送到 TxD 端。一位时间后，DATA 有效，数据由移位寄存器送入 TxD 端。再过一位后产生第一个移位脉冲。第一个移位时钟将“1”(停止位）送入移位寄存器的第 9 位，此后每次移位只把 0 送入第 9 位，所以当数据位向右移出时，“0”从左边移入。当 TB8 移至输出位置上时，它左边就是停止位，其余位均为零。此时将通知发送控制器作最后一次移位，然后使 SEND 无效并置位 TI。这些均发生在“写 SBUF”后第 11 次计数器翻转时，MCU 以 16 倍波特率对 RxD 脚进行采样，一旦检测到负跳变，16 分频计数器立即复位同时将 1FFH 写入输出移位寄存器。

在每一位的第 7、8、9 状态时，位检测器对 RxD 端值进行采样。对三个采样值取多数（至少 2 次）为确定值以抑制噪声。如若所接收的第一位不为 0，接收电路复位，单元等待下一个负跳变的出现。如果起始位有效，则被移入输入移位寄存器，并开始接收这一帧中的其它位。

数据位从右边移入，“1”从左边移出。当起始位移至寄存器（模式 2~3 时为 9 位寄存器）的最左端时，通知接收控制器进行最后一次移位，并装入 SBUF 及 RB8 并置位 RI。仅当产生最后一位移位脉冲时同时满足下列 2 个条件：①RI=0,②SM2=0 或接收到的第 9 位数据为 1 时, 才装载 SBUF 和 RB8 并置位。

上述两个条件任一不满足,所接收到的数据帧就会丢失不再恢复，RI 仍为 0。当两者都满足时，第 9 位数据位就装入 RB8，前 8 位数据则装入 SBUF，一个位时间后，无论上述条件满足与否，单元都会重新等待 RxD 端的负跳变。

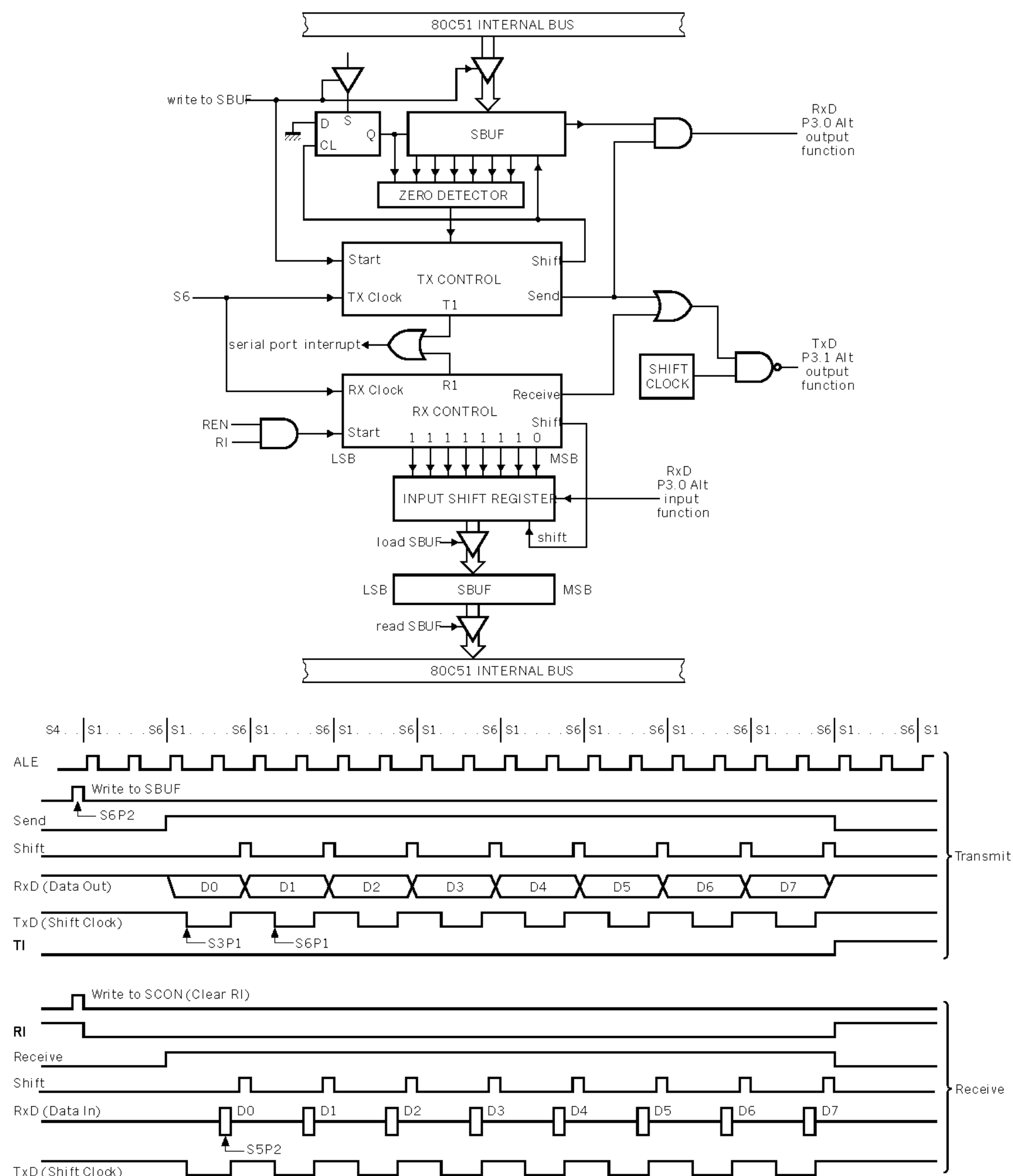

80C51 INTERNAL BUS
write to SBUF
D
S
Q
CL
SBUF
ZERO DETECTOR
RxD
P3.0 Alt
output
function
Start
TX CONTROL
Shift
S6
TX Clock
T1
Send
serial port interrupt
TxD
P3.1 Alt
output
function
SHIFT
CLOCK
RX Clock
R1
Receive
RX CONTROL
REN
RI
Start
Shift
1 1 1 1 1 1 1 0
LSB
MSB
INPUT SHIFT REGISTER
RxD
P3.0 Alt
input
function
load SBUF
shift
SBUF
read SBUF
80C51 INTERNAL BUS
ALE
Write to SBUF
S6P2
Send
Shift
RxD (Data Out)
D0
D1
D2
D3
D4
D5
D6
D7
TxD (Shift Clock)
S3P1
S6P1
TI
Transmit
Write to SCON (Clear RI)
RI
Receive
Shift
RxD (Data In)
S5P2
TxD (Shift Clock)

图 25　串口模式 0

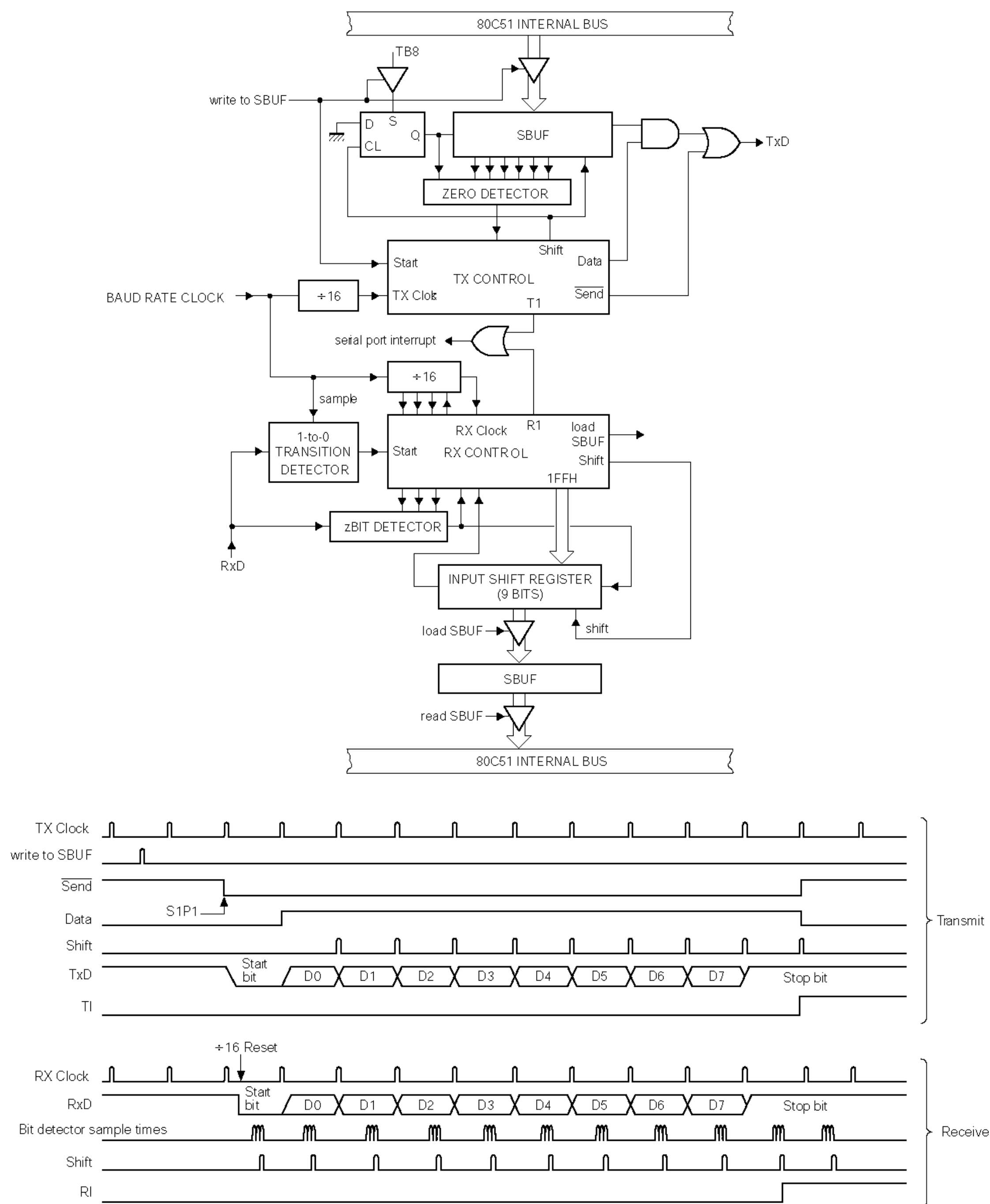

80C51 INTERNAL BUS
TB8
write to SBUF
D
S
Q
CL
SBUF
TxD
ZERO DETECTOR
Start
Shift
Data
TX CONTROL
TX Clock
T1
Send
BAUD RATE CLOCK
÷16
serial port interrupt
÷16
sample
1-to-0
TRANSITION
DETECTOR
Start
RX Clock
R1
load
SBUF
RX CONTROL
Shift
1FFH
RxD
zBIT DETECTOR
INPUT SHIFT REGISTER
(9 BITS)
load SBUF
shift
SBUF
read SBUF
80C51 INTERNAL BUS
TX Clock
write to SBUF
Send
Data
S1P1
Shift
TxD
Start bit
D0
D1
D2
D3
D4
D5
D6
D7
Stop bit
TI
Transmit
÷16 Reset
RX Clock
RxD
Start bit
Stop bit
Bit detector sample times
Shift
RI
Receive

图 26 串口模式 1

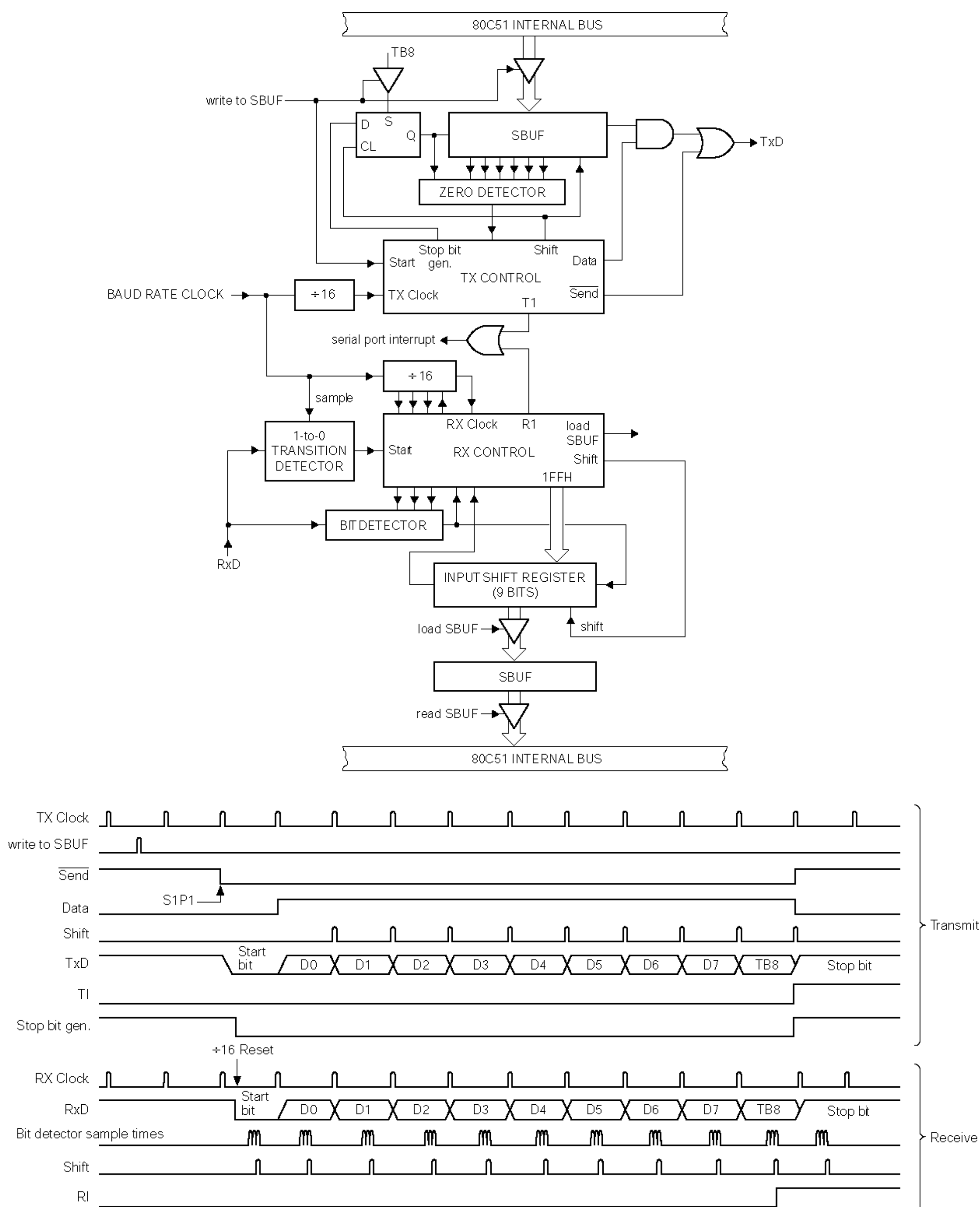


图 27 串口模式 2 和 3

14.5 增强型 UART

除了标准操作模式外，UART 可实现自动地址识别和通过查询丢失的停止位进行帧错误检测。UART 还支持多机通信。

当使用帧错误检测时，丢失的位将会置位 SCON 中的 FE 位。FE 与 SM0 共用 SCON.7，通过 PCON.6（SMOD0）选择,见表 50。如果 SMOD0 置位，SCON.7 作为 FE，SMOD0 为 0 时，SCON.7 作为 SM0。作为 FE 时，SCON.7 只能由软件清零（见图 28）。

表 49 串口控制寄存器 SOCON(地址 98H)

7	6	5	4	3	2	1	0
SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI

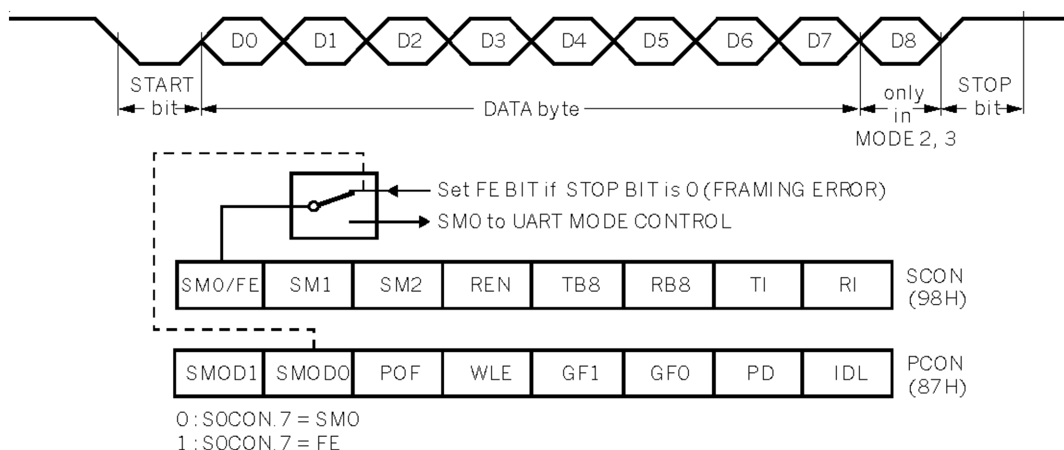


图 28　UART 帧错误检测

表 50　S0CON 位描述

位	符号	描述
7	FE SM0	帧错误位 当检测到一个无效的停止位时，该位由接收器置位。FE 位通过软件而不是有效的帧清零。 串口模式位 0（要访问 SM0，SMOD0 必须为 0），见表 46
6	SM1	用于选择串口模式，见表 46
5	SM2	使能自动地址识别　在模式 2 和 3 中，如果 SM2=1，且接收到的第 9 位数据（RB8）是 0，则 RI（接收中断标志）不会被激活。在模式 1 中，如果 SM2=1 且没有接收到有效的停止位，则 RI 不会被激活，而接收到的字节为一个给定或广播地址。在模式 0 中，SM2 必须是 0。
4	REN	使能接收。由软件置位或清除。REN=1 时，使能接收，REN=0 时，禁止接收。
3	TB8	模式 2 和 3 中发送的第 9 位数据，可以按需要由软件置位或清除。
2	RB8	模式 2 和 3 中已接收的第 9 位数据，在模式 1 中，或 SM2=0，RB8 是已接收的停止位。在模式 0 中，RB8 未用。
1	TI	发送中断标志。模式 0 中。在发送完第 8 位数据时，由硬件置位。其它模式中，在发送停止位之初，由硬件置位。在任何模式中，都必须由软件来清除 TI。
0	RI	接收中断标志，模式 0 中，接收第 8 位结束时由硬件置位。其它模式中，在接收停止位的中间时刻，由硬件置位。在任何模式(SM2 所述情况除外)必须由软件清除 RI。

14.5.1 自动地址识别

自动地址识别是这样一种特性，它使 UART 可以通过硬件比较从串行数据流中识别出特定的地址。这样就不必花费大量软件资源去检查每一个从串口输入的串行地址。将 S0CON 内 SM2 置位可使能该特性。在 9 位 UART 模式（模式 2 和模式 3）下，如果接收的字节中包含“给定”地址或“广播”地址，接收中断标志（RI）将自动置位。在 9 位模式下要求第 9 个信息位为 1 以表明该信息内容是地址而非数据，见图 29。在 8 位模式中（模式 1），如果 SM2 使能且接收的信息在 8 个地址位之后有一个有效的停止位时，RI 标志置位，接收的信息是一个给定或广播地址。模式 0 为移位寄存器模式，SM2 被忽略。

使用自动地址识别特性时，主机通过调用特定从机地址有选择地与一个（或多个）从机通信。使用广播地址时，所有从机都被联系。在此使用了两个特殊功能寄存器：SADDR 表示从机地址，SADEN 表示地址屏蔽。SADEN 用于定义 SADDR 内哪几位需使用而哪几位不予考虑。SADEN 可以与 SADDR 逻辑“与”

得出给定的地址，用于对每一从机进行寻址。示例如下：

从机 0　　SADDR=1100 0000
　　　　　SADEN=1111 1101
　　　　　特定地址=1100 00X0

从机 1　　SADDR=1100 0000
　　　　　SADEN=1111 1110
　　　　　特定地址=1100 000X

上例中 SADDR 相同，而 SADEN 不同以区分两个从机。从机 0 要求 0 位为 0 而忽略 1 位。从机 1 则要求 1 位为 0 而忽略 0 位。由于从机 1 的 1 位必须为 0，从机 0 只能取独有的地址 1100 0010 以区别。由于从机 0 的 0 位必须为 1，从机 1 只能取独有的地址 1100 0001 以区别。而取地址 1100 0000 时两从机都可被寻址。

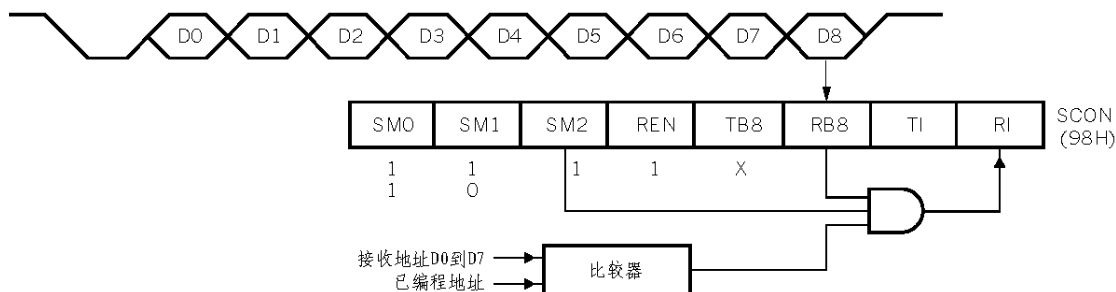
下例所示为选择从机 1、2 而不选从机 0：

从机 0　　SADDR=1100 0000
　　　　　SADEN=1111 1001
　　　　　特定地址=1100 0XX0

从机 1　　SADDR=1110 0000
　　　　　SADEN=1111 1010
　　　　　特定地址=1100 0X0X

从机 2　　SADDR=1110 0000
　　　　　SADEN=1111 1100
　　　　　特定地址=1110 00XX

上述三个从地址只有低 3 位不同。从机 0 要求位 0=0，它可通过 1110 0110 单独寻址；从机 1 要求位 1=0，可通过 1110 0101 单独寻址；从机 2 要求位 2 为 0，可通过 1110 0011 单独寻址。由于必须使地址字节的第 2 位为“1”以屏蔽从机 2，因此使用地址 1110 0100 可选通从机 0 和 1 同时屏蔽从机 2。将 SADDR 和 SADEN 相“或”后产生每个从机的“广播”地址，结果为零的位视为无关位。大多数情况下，无关位被认为是 1，这样，“广播”地址为 FFH。复位时 SADDR 和 SADEN 均为 00H，此时产生了一个所有位都是无关位的给定地址，也即“广播”地址。这样有效地禁止了自动寻址模式，并允许微控制器使用不带有上述特性的标准 UART 驱动器。



在UART Mode 2 或 Mode 3 中并且 SM2 = 1:
如果 REN = 1, RB8 = 1，则产生中断并且接收地址= 已编程地址
± 当接收到自身地址，清零SM2以接收数据字节
± 当所有数据字节都接收到时，置位SM2以等待下个地址

图 29　UART 多机通信，自动地址识别

15 SIO1，I²C 串行 IO

I²C 总线用两条线（SDA 和 SCL）在总线和装置之间传递信息。总线的主要特性如下：

- 在主器件和从器件之间采用双向数据传送方式
- 多主机总线（无中央主机）
- 多主机同时传送时进行仲裁避免总线上数据冲突
- 串行时钟的同步允许器件通过一条总线以不同的位速率进行通信
- 串行时钟的同步可用作握手机制对串行传输进行挂起或恢复
- I2C 总线可用于测试和诊断

P1.6 和 P1.7 必须设置为开漏模式（SDA 和 SCL）

P8xC591 片内 I²C 逻辑提供了符合 I²C 总线规范的串行接口。I²C 逻辑自动对字节传输进行处理，并对串行传输进行跟踪，状态寄存器（S1STA）反映了 SIO1 和 I²C 总线的状态。

CPU 和 I²C 逻辑通过下面 4 个特殊功能寄存器接口：S1CON（SIO1 控制寄存器）、S1STA（SIO1 状态寄存器）、S1DAT（SIO1 数据寄存器）和 S1ADR（SIO1 地址寄存器）。SIO1 逻辑与外部 I²C 总线通过 2 个 I/O 管脚接口：P1.6/SCL（串行时钟线）和 P1.7/SDA（串行数据线）。

一个典型的 I²C 总线配置见图 30，而图 31 所示为数据传输在总线上是如何实现的。根据方向位（R/W）的状态，在 I²C 总线上有两种数据传输类型：

1. 从主机发送器到从机接收器的数据传输。主机发送的第一个字节为从机地址，后面是数据字节。从机每接收到一个字节返回一个应答位。
2. 从从机发送器到主机接收器的数据传输。第一个字节（从地址）由主机发送，然后从机返回一个应答位。接下来是从机向主机发送的数据字节。主机每接收一个字节会返回一个应答位，但最后一个字节除外。接收最后一个字节后不会返回应答位。

主机产生所有的串行时钟和起始以及停止条件。传输以一个停止条件或一个重复的起始条件结束。由于重复的起始条件是下一次传输的开始，因此 I²C 总线不会被释放。

15.1 操作模式

片内 SIO1 逻辑具有下列 4 种操作模式：

1. 主发送器模式： 串行数据通过 P1.7/SDA 输出而串行时钟通过 P1.6/SCL 输出。发送的第一个字节包括接收器件的从地址（7 位）和数据方向位。在该模式中数据方向位为逻辑 0，即传输的是“W”，这样第一个发送的字节为 SLA+W。串行数据一次发送 8 位。每发送一个字节接收一个应答位。起始和停止条件用于指示串行传输的开始和结束。
2. 主接收器模式： 发送的第一个字节包括接收器件的从地址（7 位）和数据方向位。在该模式中数据方向位为逻辑 1，即传输的是“R”，这样第一个发送的字节为 SLA+R。串行数据一次接收 8 位。每接收一个字节发送一个应答位。起始和停止条件用于指示串行传输的开始和结束。
3. 从接收器模式： 串行数据和串行时钟通过 P1.7/SDA 和 P1.6/SCL 接收。每接收一个字节发送一个应答位。起始和停止条件用于指示串行传输的开始和结束。在接收从地址和方向位之后，通过硬件实现地址识别。
4. 从发送器模式： 第一个字节的接收和处理与从接收器模式相同。但在该模式中，方向位指示发送方向是相反的。串行数据通过 P1.7/SDA 输出而串行时钟通过 P1.6/SCL 输入。起始和停止条件用于指示串行传输的开始和结束。

在一个给定的应用中，SIO1 可作为主机或者从机。在从模式中，SIO1 的硬件寻找自己的从地址和通用的调用地址。如果检测到其中一个，请求一个中断。当微控制器想要成为总线的主机时，进入主模式之前硬件一直等到总线释放，这样就不会中断一个可能的从动作。如果在主模式中丢失总线仲裁，SIO1 立即切换到从模式并可在同一个串行传输中检测到自己的从地址。

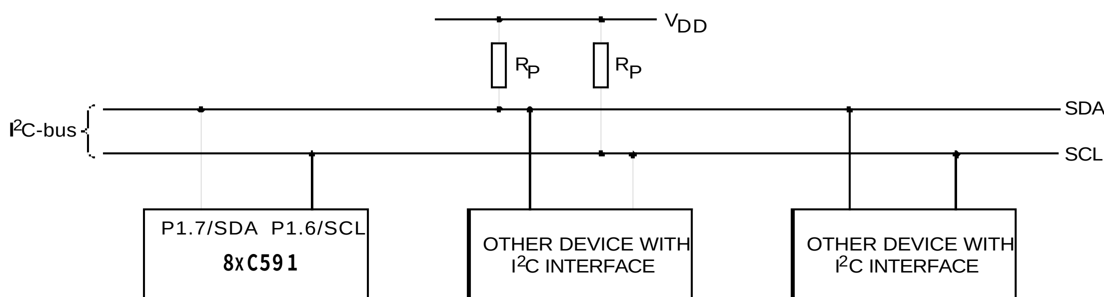


图 30 典型的 I^2C 总线配置

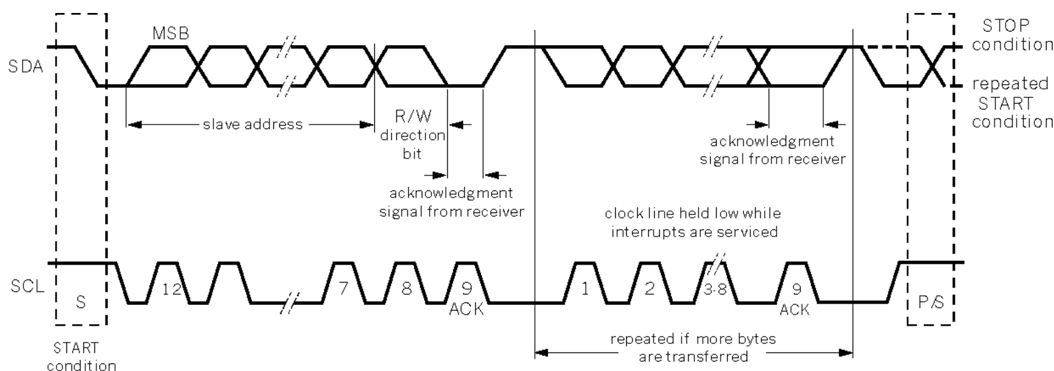


图 31 I^2C 总线上的数据传输

15.2 SIO1 的执行和操作

图 32 所示为片内 I^2C 总线接口如何执行。下面的章节讲述独立的模块

15.2.1 输入滤波器和输出部分

输入滤波器兼容 I^2C 输入电平。如果输入电压小于 1.5V，输入逻辑电平被认为是 0；如果输入电压高于 3.0V，输入逻辑电平被认为是 1。输入信号和内部时钟（$f_{CLK}/4$）同步，小于 3 个振荡器周期的尖峰被滤除。

输出部分包括开漏晶体管，在 Vout<0.4V 时可驱动 3mA。这些开漏输出有连接到 V_{DD} 的钳位二极管。如果掉电的 PP8xC591 将 I^2C 总线外部钳位，必须考虑到预防措施。

15.2.2 地址寄存器，S1ADR

该 8 位特殊功能寄存器在编程为从发送器或接收器时可装入 SIO1 所响应的 7 位从地址（高 7 位）。最低位 GC 用于使能通用调用地址（00H）识别。

15.2.3 比较器

比较器用自身的从地址（S1ADR 中的高 7 位）与接收到的 7 位从地址进行比较。它还比较用通用调用地址（00H）与接收的第一个 8 位字节比较。如果相等，对应的状态位置位并请求中断。

15.2.4 移位寄存器，S1DAT

该 8 位特殊功能寄存器包含一个要发送的串行数据字节或一个以接收的字节。S1DAT 中的数据总是从右向左移位。发送的第一个位是最高位（Bit7），而在接收完一个字节后，第一个接收数据位是 S1DAT 的最高位。数据移出时，总线上的数据同时移入。S1DAT 总是包含当前总线上的最后一个字节。因此在丢失仲裁的情况下，主发送器连同 S1DAT 中的正确数据转换到从接收器。

图 32 I^2C 总线接口方框图

15.2.5 仲裁和同步逻辑

在主发送器模式中，仲裁逻辑检测每个发送的逻辑 1 实际在 I^2C 总线上表现为逻辑 1。如果总线上另一个器件否决了逻辑 1 并将 SDA 线拉低，仲裁将会丢失，而 SIO1 立即从主发送器变为从接收器。SIO1 将继续输出时钟脉冲（SCL 上）直到当前串行字节发送完毕。

仲裁也可能在主接收器模式中丢失。该模式中的仲裁丢失只会发生在 SIO1 向总线返回一个非应答位（逻辑 1）时。当总线上另一个器件将该信号拉低时，仲裁丢失。由于这只能发生在串行字节的结束，SIO1 不再产生时钟脉冲。图 33 所示为仲裁过程。

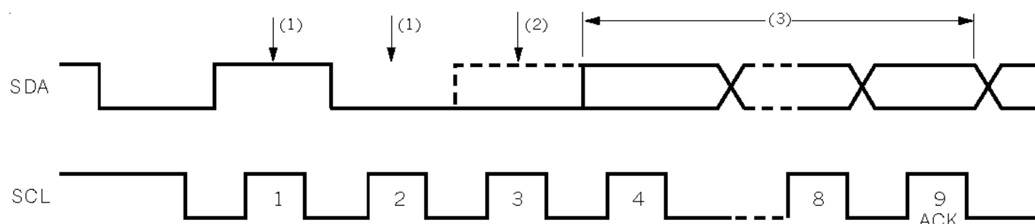


图 33 仲裁过程

同步逻辑将串行时钟发生器与 SCL 上另一个器件的时钟脉冲同步。如果两个或更多的主器件产生时钟脉冲，高电平的宽度由产生最短高电平宽度的器件决定，而低电平宽度由产生最长低电平宽度的器件决定。图 34 所示为同步过程。

从器件可延长低电平宽度以使总线主机减慢。用于握手时也可延长低电平宽度。可在每个位或一个完整字节传输后实现。SIO1 将在发送或接收一个字节并且发送应答位之后延长 SCL 低电平宽度。串行中断标志（SI）置位且延长持续到串行中断标志清零。

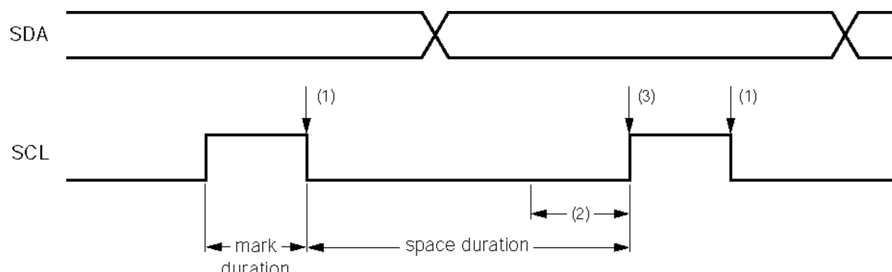


图 34　串行时钟同步

15.2.6　串行时钟发生器

该可编程时钟脉冲发生器当 SIO1 处于主发送器模式或主接收器模式时提供 SCL 时钟脉冲。可编程的输出时钟频率有：f_{CLK}/120、f_{CLK}/9600 和 8 分频后的定时器 1 溢出速率。输出时钟脉冲有 50%占空比（时钟发生器与其它时钟源同步时除外）。

15.2.7　定时和控制

定时和控制逻辑产生用于串行字节处理的定时和控制信号。该逻辑模块提供 S1DAT 的移位脉冲，使能比较器，产生和检测起始和停止条件，接收和发送应答位，控制主和从模式，包含中断请求逻辑以及监控 I^2C 总线状态。

15.2.8　控制寄存器，S1CON

该 7 位特殊功能寄存器由微控制器使用，用于控制下面的 SIO1 功能：启动和重新启动串行传输，终止串行传输，位速率，地址识别和应答。

15.2.9　状态解码器和状态寄存器

状态解码器提取所有的内部状态位并将它们压缩成 5 位代码。该代码对每个 I^2C 总线状态都是唯一的。该 5 位代码可用于产生向量地址用于不同服务程序的快速处理。每个服务程序处理一个特定的总线状态。如果使用了 SIO1 的所有 4 种模式，就有 26 种可能的总线状态。当串行中断标志置位（由硬件）时，5 位状态代码被锁存到状态寄存器的高 5 位，并在中断标志由软件清零之前保持不变。状态寄存器的低 3 位总是为 0。如果状态代码作为服务程序的入口向量，那么服务程序将按照每隔 8 个地址位置分配。8 个字节代码对大多数服务程序来说是足够的（见该章的软件示例）。

15.2.10　4 个 SIO1 特殊功能寄存器

微控制器通过 4 个特殊功能寄存器与 SIO1 接口，这 4 个 SFR（S1ADR，S1DAT，S1CON 和 S1STA）在下面的章节中分别讲述。

15.2.10.1　地址寄存器，S1ADR

CPU 可对该 8 位直接寻址寄存器进行读写。S1ADR 不受 SIO1 硬件影响。当 SIO1 为主模式时，该寄存器的内容无关。在从模式中，高 7 位必须装入微控制器自身的从地址，并且当最低位置位时通用调用地址（00H）被识别，否则忽略。

在起始条件之后，最高位对应于从 I^2C 总线接收到的第一个位。S1ADR 中的逻辑 1 对应 I^2C 总线上的高电平。逻辑 0 对应总线上的低电平。

表 51　地址寄存器 S1ADR（地址 DBH）

7	6	5	4	3	2	1	0
X	X	X	X	X	X	X	GC

表 52 S1ADR 位描述

位	符号	描述
7～1	X	自身从地址
0	GC	0＝不识别通用调用地址 1=识别通用调用地址

15.2.11 数据寄存器，S1DAT

S1DAT 包含一个要发送的串行数据字节或一个刚接收到的字节。当该 8 位直接寻址寄存器没有处理移位数据时，CPU 可对其进行读写。这发生在 SIO1 处于已定义的状态且串行中断标志置位时。S1DAT 中的数据在 SI 置位时一直不变。S1DAT 中的数据总是从右向左移位：发送的第一个位是最高位（Bit7），而在接收完一个字节后，接收到的第一个数据位是 S1DAT 的最高位。当数据移出时，总线上的数据同时移入；S1DAT 总是包含当前总线上的最后一个数据字节。因此在丢失仲裁的情况下，主发送器连同 S1DAT 中的正确数据转换到从接收器。

S1DAT 和 ACK 标志组成一个 9 位移位寄存器，将 8 位字节连同后面的应答位移入移出。ACK 标志由 SIO1 硬件控制，不能通过 CPU 访问。串行数据通过 ACK 标志在 SCL 线上的串行时钟脉冲上升沿移入 S1DAT。当一个字节完全移入 S1DAT，则 S1DAT 中的数据可被访问。在第 9 个时钟脉冲由硬件产生应答位。串行数据通过一个缓冲器（BSD7）在 SCL 线的时钟脉冲下降沿从 S1DAT 移出。当 CPU 写入 S1DAT 时，BSD7 装入 S1DAT.7 的内容，这是要发送到 SDA 线的第一个位（见图 36）。在 9 个串行时钟脉冲之后，S1DAT 的 8 个位发送到 SDA 而 ACK 中的应答位即将发送。注意，8 个已发送的位已经移回到 S1DAT。

表 53 数据寄存器 S1DAT（地址 DAH）

7	6	5	4	3	2	1	0
SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0

表 54 S1ADR 位描述

位	符号	描述
7~0	SD7~SD0	要发送的或刚接收的 8 个位。S1DAT 中的逻辑 1 对应 I^2C 总线上的高电平。逻辑 0 对应总线上的低电平。串行数据通过 S1DAT 从右到左移位。图 35 所示为 S1DAT 中的数据是如何 SDA 进行传输的。

15.2.12 控制寄存器，S1CON

CPU 可对该 8 位直接寻址寄存器进行读写。其中两个位受硬件的影响：当请求串行中断时 SI 位置位，当 I^2C 总线上出现一个停止条件时 STO 位清零。当 ENS1=0 时 STO 位也清零。

表 55 控制寄存器 S1CON（地址 D8H）

7	6	5	4	3	2	1	0
CR2	ENS1	STA	STO	SI	AA	CR1	CR0

表 56 S1CON 位描述

位	符号	描述
7	CR2	时钟速率位 2 见表 57
6	ENS1	使能串行口 ENS1=0：I^2C 禁止和复位。ENS1=1：串行口使能
5	STA	起始标志 当该位在从模式中置位，硬件检测 I^2C 总线，如果总线空闲或等到总线空闲后产生一个起始条件。如果器件处于主模式中，将产生一个重复的起始条件。
4	STO	停止标志 如果该位在主模式中置位则产生一个停止条件。I^2C 总线检测到的停止条件将清零该位。该位还可在从模式中置位以使总线从错误条件中恢复。该情况下，不对 I^2C 总线产生停止条件，而是由硬件释放 SDA 和 SCL 线并切换到没有选择的接收器模式。停止标志由硬件清零。
3	SI	串行中断标志 下列事件中的任何一个发生后，该标志置位并产生一个中断请求： • 主模式中产生一个起始条件 • 当 AA=1 时接收到自身的从地址 • 当 S1ADR.0 和 AA=1 时接收到通用调用地址 • 主模式中接收或发送一个数据字节（即使仲裁丢失） • 所选择的从模式接收或发送一个数据字节 • 所选择的从接收器或发送器接收到一个停止或起始条件 当 SI 标志置位，SCL 保持低电平且串行发送被挂起时，SI 必须由软件复位
2	AA	声明应答标志 当该位置位时，下面的任意条件之一将产生一个应答： • 接收到自身的从地址 • 接收到通用调用地址（S1ADR.0=1） • 当器件编程为一个主接收器时，接收到一个数据字节 • 当器件为所选择的从接收器时，接收到一个数据字节 当该位复位时，不返回应答标志。因此，当接收到自身地址或通用调用地址时不会产生中断请求。
1	CR1	时钟速率位 1 和 0；见表 57
0	CR0	

15.2.12.1 ENS1，SIO1 使能位

ENS1=0：当 ENS1 为 0 时，SDA 和 SCL 输入信号被忽略，SIO1 处于一个非寻址的从状态，而 S1CON 中的 STO 位强制为 0。其它位不受影响。

ENS1=1：当 ENS1 为 1 时，I^2C 使能。注意，P1.6 和 P1.7 必须通过将口模式寄存器 P1M1.x 和 P1M2.x 位 6 和 7 写为 1 设置为开漏。（见 6.2 管脚描述）

ENS1 不能用于暂时从 I^2C 总线释放 SIO1，因为当 ENS1 复位时 I^2C 总线的状态丢失。AA 标志可用于暂时释放 SIO1(见后面 AA 标志的描述)。

15.2.12.2 STA，起始标志

STA=1： 当 STA 位置位以进入主模式时，SIO1 硬件检测 I^2C 总线的状态。如果 I^2C 总线空闲就产生一个起始条件。如果总线没有空闲，SIO1 等待一个停止条件（释放总线）并在内部串行时钟发生器的半个时钟周期之后产生一个起始条件。

如果 STA 置位时 SIO1 已经处于主模式并且发送或接收了一个或多个字节,SIO1 发送一个重复的起始条件。STA 在任何时候都可置位。当 SIO1 处于被寻址的从模式时也可置位 STA。

STA=0： 当 STA 复位后，将不会产生起始条件或重复起始条件。

15.2.12.3 STO，停止标志

STO=1：当 STO 置位而 SIO1 处于主模式时，向 I^2C 总线发送停止条件。当检测到总线上的停止条件时，SIO1 硬件清零 STO 标志。在从模式中，STO 标志可置位以从错误条件中恢复。此情况下不向 I^2C 总线发送停止条件。但是 SIO1 的硬件表现为好像接收到停止条件并切换到已定义的非寻址的从接收器模式。STO 标志由硬件自动清零。

如果 STA 和 STO 同时置位，SIO1 处于主模式时向 I^2C 总线发送停止条件（从模式时 SIO1 产生一个不被发送的内部停止条件）。然后发送一个起始条件。

STO=0： 当 STO 复位时，不产生停止条件。

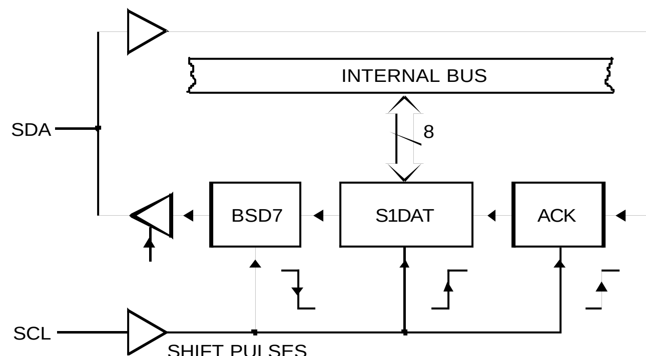


图 35 串行输入/输出配置

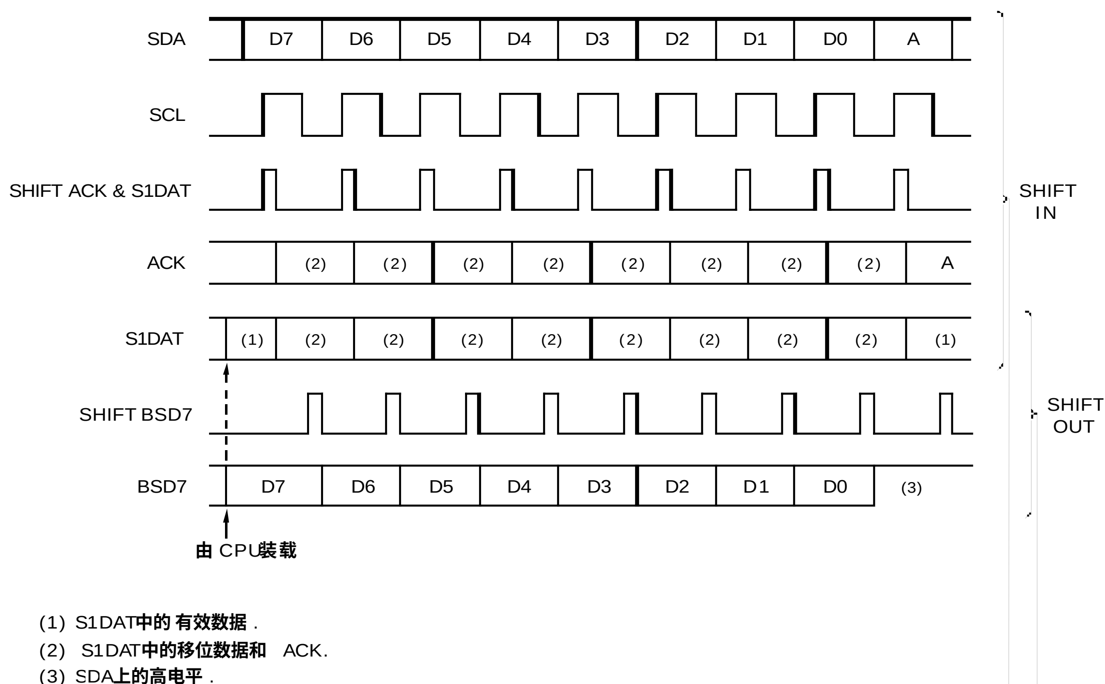


图 36 移入和移出时序

15.2.12.4 SI，串行中断标志

SI=1： 当 SI 标志置位且 EA 和 ES1（中断使能寄存器）位都置位时，产生一个串行中断请求。当进入 26 个可能的 SIO1 状态中的 25 个其中之一时，SI 由硬件置位。唯一不使 SI 置位的状态是 F8H，它指示没有可用的相关状态信息。当 SI 置位时，SCL 的串行时钟低电平周期被延长，而串行传输被挂起。SCL 的高电平周期不受串行中断标志影响。SI 必须由软件复位。

SI=0： 当 SI 标志复位后，不再产生串行中断请求，SCL 的串行时钟不再延长。

15.2.12.5 AA，声明应答标志

AA=1： 如果 AA 标志置位，下列条件之一将在 SCL 上的应答时钟脉冲时返回一个应答位（SDA 的低电平）：

- 接收到自身的从地址
- 当 S1ADR 中的 GC 位置位时接收到通用调用地址（S1ADR.0=1）
- 当 SIO1 处于主接收器模式时，接收到一个数据字节
- 当 SIO1 处于被寻址的从接收器模式时，接收到一个数据字节

AA=0： 如果 AA 标志复位，下列条件之一将在 SCL 上的应答时钟脉冲时返回一个非应答位（SDA 的高电平）：

- 当 SIO1 处于主接收器模式时，接收到一个数据字节
- 当 SIO1 处于被寻址的从接收器模式时，接收到一个数据字节

当 SIO1 处于被寻址的从发送器模式时，发送最后一个串行数据后进入状态 C8H（见图 40）。当 SI 清零时，SIO1 退出 C8H，进入非寻址的从接收器模式，而 SDA 线保持高电平。在状态 C8H 中，AA 标志可再次置位用于以后的地址识别。

当 SIO1 处于非寻址的从模式时，自身的从地址和通用调用地址被忽略。因此不会返回应答位，也不会产生串行中断请求。这样 SIO1 可暂时从 I^{2}C 总线上释放而保持对总线状态的监控。SIO1 从总线释放期间，起始和停止条件被检测并且串行数据被移入。通过置位 AA 标志可随时恢复地址识别。如果当自身从地址或通用调用地址被部分接收时置位 AA 标志，在字节发送结束时地址将被识别。

15.2.12.6 CR0,CR1 和 CR2，时钟速率位

这 3 个位决定了 SIO1 处于主模式时的串行时钟频率。可变的串行时钟速率见表 57。器件可使用 12.5KHz 的位速率通过软件驱动的标准 I/O 口与 I^{2}C 总线接口。100KHz 通常是最大的位速率，可由 16MHz，12MHz 或一个 6MHz 振荡器产生。当 SIO1 处于主模式时，如果定时器 1 不作其它用途，可使用可变的位速率（0.5KHz 到 62.5KHz）。表 57 所示的频率在 SIO1 处于从模式时并不重要。在从模式中，SIO1 自动与最大为 100KHz 的时钟频率同步。

15.2.13 状态寄存器，S1STA

S1STA 是一个 8 位只读特殊功能寄存器。低 3 位总是为 0。高 5 位包含状态代码。有 26 个可能的状态。当 S1STA 包含 F8H 时表示无可用的相关状态信息，也没有串行中断请求。S1STA 所有其它值都对应一个已定义的 SIO1 状态。进入每个状态都会产生串行中断请求（SI=1）。SI 置位一个机器周期后当前 S1STA 中的代码有效。在 SI 由软件复位的一个机器周期之后，此代码仍然存在。

表 57 串行时钟速率

CR2	CR1	CR0	位速率(KHz)			分频系数
			6MHz	8MHz	12MHz	
0	0	0	47	62.5	94	128
0	0	1	54	71	107[1]	112
0	1	0	63	83.3	125[1]	96
0	1	1	75	100	150[1]	80
1	0	0	12.5	17	25	480
1	0	1	100	133[1]	200[1]	60
1	1	0	200[1]	267[1]	400[1]	30
1	1	1	0.49>62.5 0<254	0.65<55.6 0<253	0.98<50.0 0<251	48×(256(定时器 1 重装值)) 定时器 1 在模式 2 中的重装值

注 1：这些频率超过了标准 I^2C 总线规范的 100KHz 上限。

15.2.14 有关 SIO1 操作模式更多的信息

4 种操作模式分别为：

- 主发送器
- 主接收器
- 从接收器
- 从发送器

每种操作模式的数据传输如图 37~40 所示。这些图中包含了以下缩写：

缩写	解释
S	起始条件
SLA	7 位从地址
R	读位（SDA 上高电平）
W	写位（SDA 上低电平）
A	应答位（SDA 上低电平）
$\overline{A}$	非应答位（SDA 上高电平）
Data	8 位数据字节
P	停止条件

在图 37~40 中，圆圈用于指示串行中断标志位置位的时刻。圆圈中的数字显示 S1STA 寄存器中的状态代码。在这些点必须执行服务程序以继续或完成串行传输。由于串行中断一直挂起到串行中断标志位由软件清零时，因此这些服务程序并不关键。

当进入一个串行中断服务程序时，S1STA 中的状态代码用于转移到适当的服务程序。对每个状态代码，随后的串行传输所需要的软件动作和细节由表 61 到 65 给出。

15.2.14.1 主发送器模式

在主发送器模式中，将一些数据字节发送到一个从接收器（见图 37）。在进入主发送器模式之前，S1CON 必须按照表 58 进行初始化。

CR0，CR1 和 CR2 定义串行位速率。ENS1 必须置位以使能 SIO1。如果 AA 位复位，而另一个器件成为总线的主控器时，SIO1 将不会对它自身的从地址或通用调用地址产生应答。换句话说，如果 AA 复位，SIO1 不能进入从模式。STA，STO 和 SI 必须复位。

主发送器模式初始化之后，可使用 SETB 指令置位 STA 进入主发送器模式。SIO1 逻辑对 I^2C 总线进行测试并在总线空闲后立即产生一个起始条件。当发送起始条件时，串行中断标志 SI 置位，状态寄存器（S1STA）中的状态代码为 08H。该状态代码用于指向一个中断服务程序。该中断程序将从地址和数据方向位（SLA+W）装入 S1DAT。S1CON 中的 SI 位必须在串行传输可以继续进行之前复位。

当从地址和方向位已发送且接收到应答位之后，串行中断标志 SI 再次置位并且可以使用 S1STA 中的一些状态代码。18H，20H 或 38H 用于主模式，68H，78H 或 B0H 用于从模式（AA=1）。每个状态代码适当的执行动作见表 61。在一个重复起始条件（状态 10H）之后，SIO1 可通过将 SLA+R 装入 S1DAT 切换到主接收器模式。

表 58 地址寄存器 S1CON（地址 D8H）

7	6	5	4	3	2	1	0
CR2	ENS1	STA	STO	SI	AA	CR1	CR0
位速率	1	0	0	0	X	位速率	

15.2.14.2 主接收器模式

在主接收器模式中，从从发送器接收一些数据字节（见图 38）。在主发送器模式时对传输初始化。发送完起始条件后，中断服务程序必须将 7 位从地址和数据方向位（SLA+R）装入 S1DAT。S1CON 中的 SI 位必须在串行传输可以继续进行之前清零。

当从地址和方向位已发送且接收到应答位之后，串行中断标志 SI 再次置位并且可以使用 S1STA 中的一些状态代码。40H，48H 或 38H 用于主模式，68H，78H 或 B0H 用于从模式（AA=1）。每个状态代码适当的执行动作见表 62。ENS1，CR1 和 CR0 不受串行传输的影响且没有在表 62 中被提到。在一个重复起始条件（状态 10H）之后，SIO1 可通过将 SLA+W 装入 S1DAT 切换到主发送器模式。

15.2.14.3 从接收器模式

在从接收器模式中，从主发送器接收一些数据字节（见图 39）。要初始化从接收器模式，S1ADR 和 S1CON 必须按照表 59 装载。高 7 位为 SIO1 被主控器寻址时所响应的地址。如果 LSB（GC）置位，SIO1 将响应通用调用地址（00H）。GC 为 0 则忽略通用调用地址。

CR0，CR1 和 CR2 在从模式中不影响 SIO1。ENS1 必须置位以使能 SIO1。AA 位必须使能以使 SIO1 应答自身的从地址或通用调用地址。STA，STO 和 SI 必须复位。

当 S1ADR 和 S1CON 完成初始化时，SIO1 等待直到它被自身的从地址加上数据方向位（必须为 0）寻址。这样使 SIO1 操作在从接收器模式中。在接收到自身从地址和 W 位后，串行中断标志（SI）置位并可从 S1STA 中读出一个有效的状态代码。该状态代码用于指向一个中断服务程序，每个代码适当的执行动作见表 63。如果仲裁丢失但 SIO1 处于主模式，也可以进入从接收器模式（见状态 68H 和 78H）。

如果在传输中 AA 位复位，SIO1 将在接收下个数据字节后向 SDA 返回一个非应答信号（逻辑 1）。当 AA 复位后，SIO1 不响应自身的从地址或通用调用地址。但 I^2C 总线仍然被监控，而地址识别可随时通过置位 AA 而恢复。这意味着 AA 位可用于暂时将 SIO1 与 I^2C 总线隔离。

表 59 地址寄存器 S1ADR（DBH）（地址 00H）

7	6	5	4	3	2	1	0
X	X	X	X	X	X	X	GC
自身从地址							

表 60 地址寄存器 S1CON（D8H）（地址 00H）

7	6	5	4	3	2	1	0
CR2	ENS1	STA	STO	SI	AA	CR1	CR0
X	1	0	0	0	1	X	X

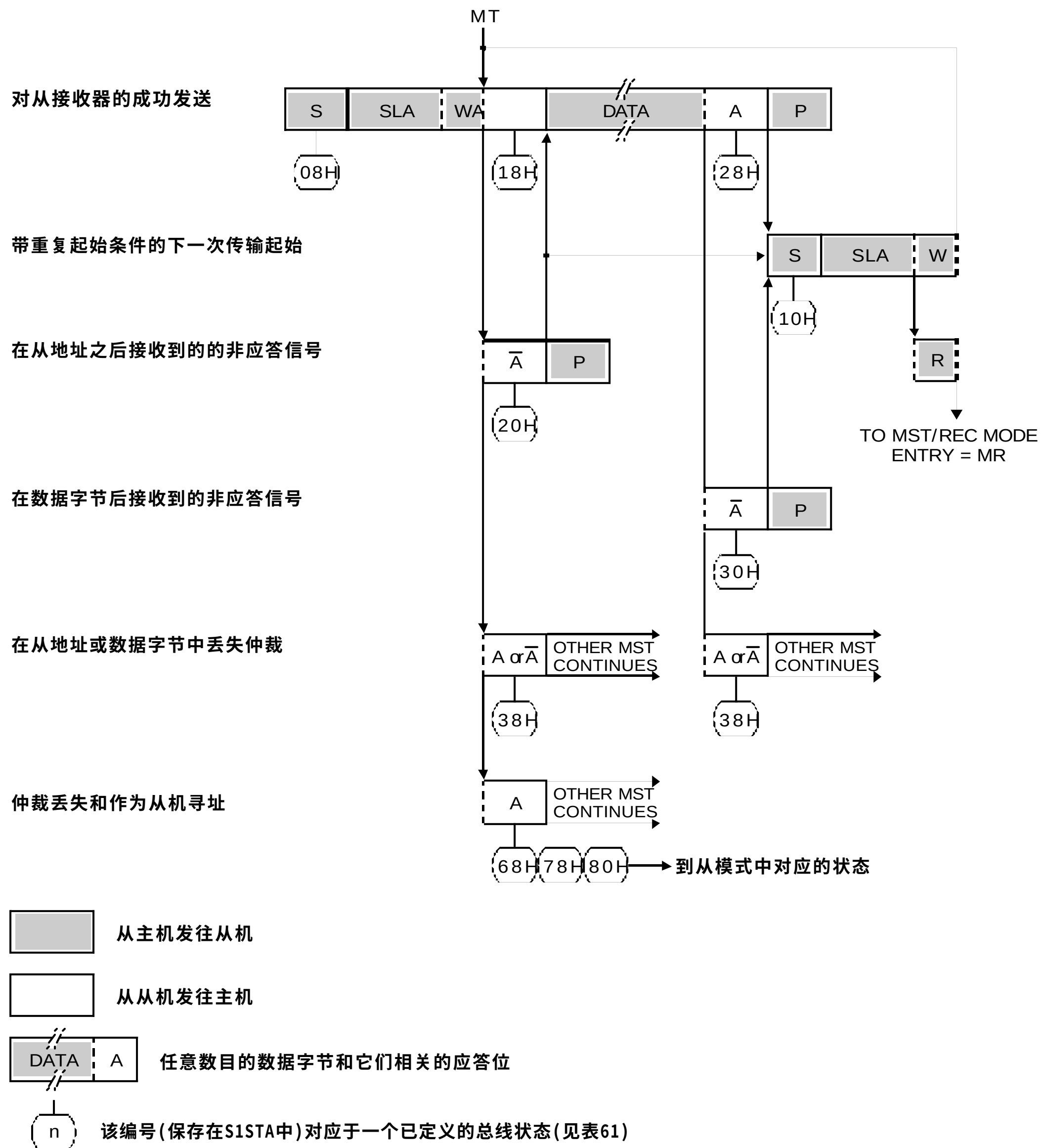


图 37 主发送器模式中的格式和状态

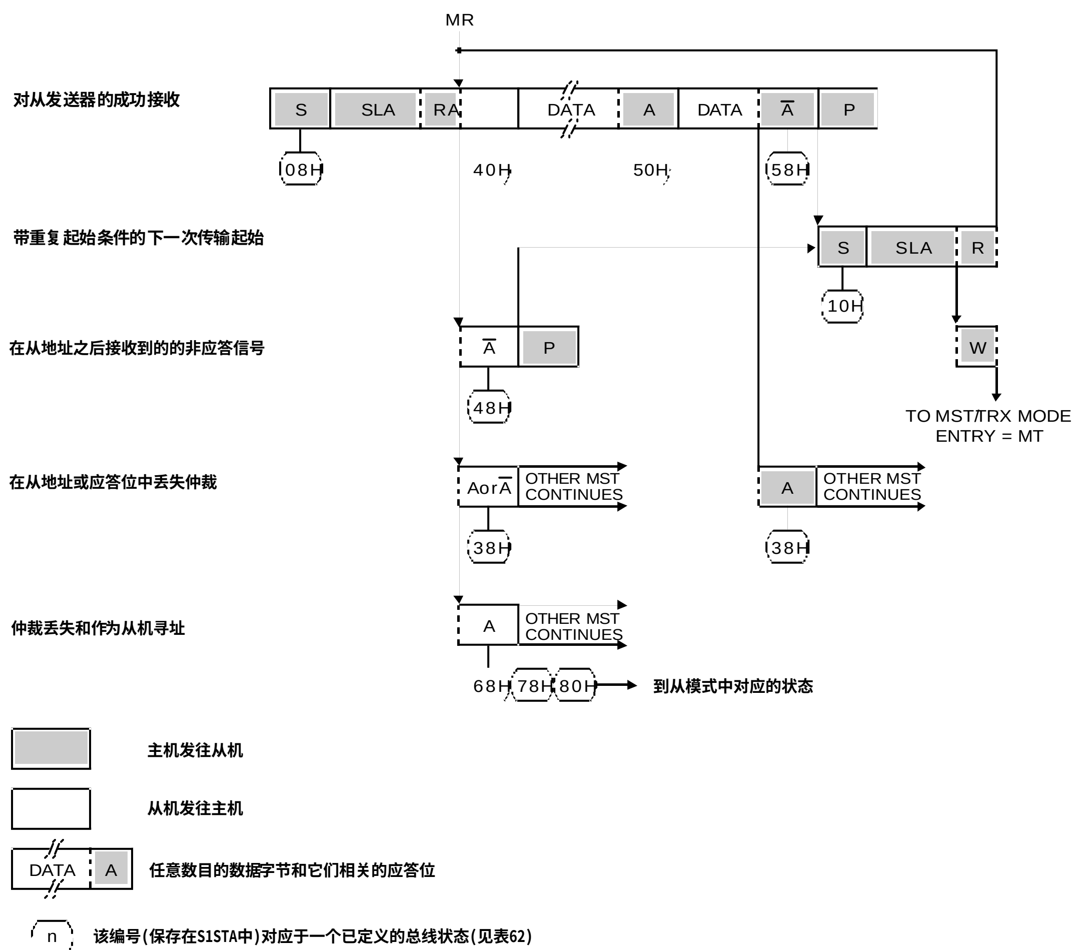


图 38　主接收器模式中的格式和状态

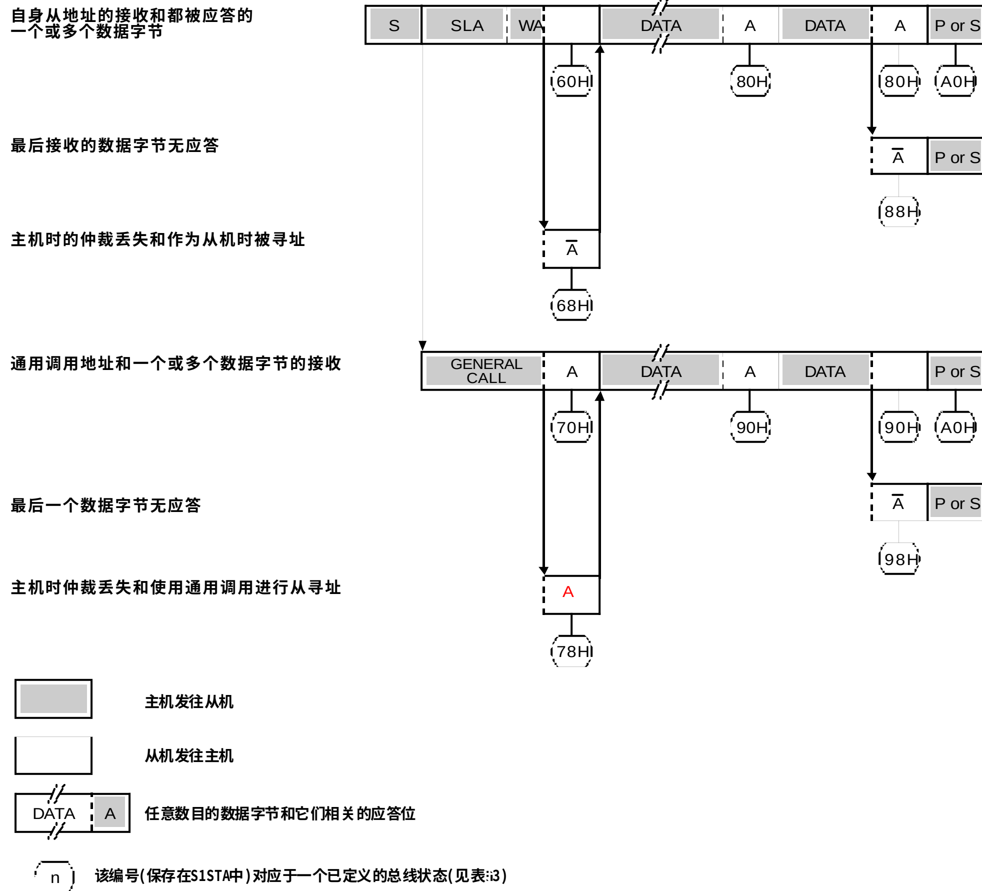


图 39　从接收器模式中的格式和状态

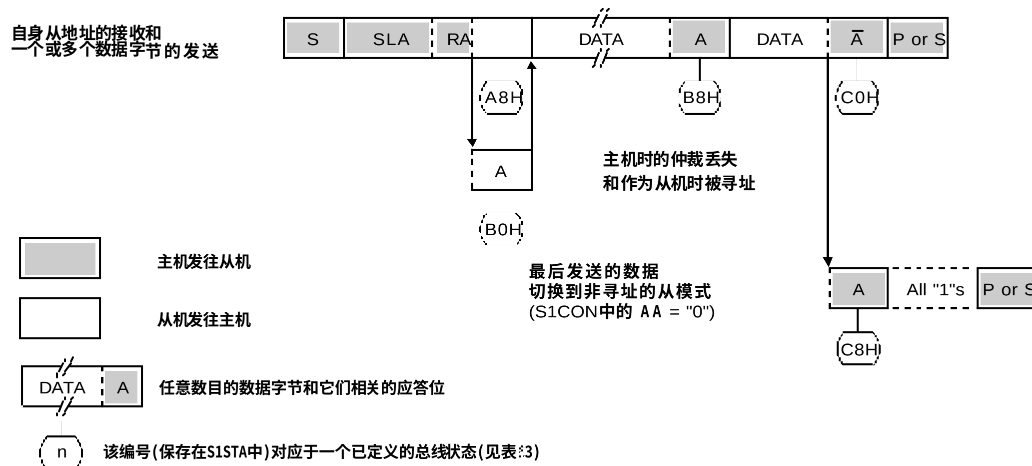


图 40　从发送器模式中的格式和状态

表 61 主发送器模式

状态码 (S1STA)	I^2C 总线及 SIO1 硬件状态	应用软件的响应					SIO1 硬件执行的下一个动作
		读/写 S1DAT	写 S1CON				
			STA	STO	SI	AA	
08H	已发送起始条件	装入 SLA+W	X	0	0	X	将发送 SLA+W，接收 ACK 位
10H	已发送重复起始条件	装入 SLA+W 或	X	0	0	X	同上
		装入 SLA+R	X	0	0	X	将发送 SLA+W，SIO1 将切换到 MST/REC 模式
18H	已发送 SLA+W；已接收 ACK	装入数据字节或	0	0	0	X	将发送数据字节，接收 ACK 位
		无 S1DAT 动作或	1	0	0	X	将发送重复起始条件
		无 S1DAT 动作或	0	1	0	X	将发送停止条件；STO 标志将复位
		无 S1DAT 动作	1	1	0	X	将发送跟在起始条件之后的停止条件；STO 标志将复位
20H	已发送 SLA+W；已接收非 ACK	装入数据字节或	0	0	0	X	将发送数据字节，接收 ACK 位
		无 S1DAT 动作或	1	0	0	X	将发送重复起始条件
		无 S1DAT 动作或	0	1	0	X	将发送停止条件；STO 标志将复位
		无 S1DAT 动作	1	1	0	X	将发送跟在起始条件之后的停止条件；STO 标志将复位
28H	已发送 S1DAT 中的数据字节；已接收 ACK	装入数据字节或	0	0	0	X	将发送数据字节，接收 ACK 位
		无 S1DAT 动作或	1	0	0	X	将发送重复起始条件
		无 S1DAT 动作或	0	1	0	X	将发送停止条件；STO 标志将复位
		无 S1DAT 动作	1	1	0	X	将发送跟在起始条件之后的停止条件；STO 标志将复位
30H	已发送 S1DAT 中的数据字节；已接收非 ACK	装入数据字节或	0	0	0	X	将发送数据字节，接收 ACK 位
		无 S1DAT 动作或	1	0	0	X	将发送重复起始条件
		无 S1DAT 动作或	0	1	0	X	将发送停止条件；STO 标志将复位
		无 S1DAT 动作	1	1	0	X	将发送跟在起始条件之后的停止条件；STO 标志将复位
38H	在 SLA+R/W 或数据字节中丢失仲裁位	装入数据字节或	0	0	0	X	I^2C 总线将被释放；进入非寻址从模式
		无 S1DAT 动作	1	0	0	X	当总线变为空闲时发送起始条件

表 62　主接收器模式

状态码 (S1STA)	I^2C 总线及 SI01 硬件状态	应用软件的响应 读/写 S1DAT	写 S1CON STA	STO	SI	AA	SI01 硬件执行的下一个动作
08H	已发送起始条件	装入 SLA+W	X	0	0	X	将发送 SLA+R，接收 ACK 位
10H	已发送重复起始条件	装入 SLA+R 或	X	0	0	X	同上
		装入 SLA+W	X	0	0	X	将发送 SLA+W,SIO1 将切换到 MST/TRX 模式
38H	在非 ACK 位中丢失仲裁	无 S1DAT 动作或	0	0	0	X	I^2C 总线将被释放；SIO1 将进入从模式
		无 S1DAT 动作	1	0	0	X	当总线恢复空闲后发送起始条件
40H	已发送 SLA+R；已接收 ACK	无 S1DAT 动作或	0	0	0	0	将接收数据字节；返回非 ACK 位
		无 S1DAT 动作	0	0	0	1	将接收数据字节；返回 ACK 位
48H	已发送 SLA+R；已接收非 ACK	无 S1DAT 动作或	1	0	0	X	将发送重复起始条件
		无 S1DAT 动作或	0	1	0	X	将发送停止条件；STO 标志将复位
		无 S1DAT 动作	1	1	0	X	将发送跟在起始条件之后的停止条件；STO 标志将复位
50H	已接收数据字节；已返回非 ACK	读数据字节或	0	0	0	0	将接收数据字节，返回非 ACK 位
		读数据字节	0	0	0	1	将接收数据字节；返回 ACK 位
58H	已接收数据字节；已返回 ACK	读数据字节或	1	0	0	X	将发送重复起始条件
		读数据字节或	0	1	0	X	将发送停止条件；STO 标志将复位
		读数据字节	1	1	0	X	将发送跟在起始条件之后的停止条件；STO 标志将复位

表 63 从接收器模式

状态码 (S1STA)	I^2C 总线及 S101 硬件状态	应用软件的响应					S101 硬件执行的下一个动作
		读/写 S1DAT	写 S1CON				
			STA	STO	SI	AA	
60H	已接收自身 SLA+W;已返回 ACK	无 S1DAT 动作或	X	0	0	0	将接收数据字节并返回非 ACK 位
		无 S1DAT 动作	X	0	0	1	将接收数据字节并返回 ACK 位
68H	主控器时在 SLA+W 中丢失仲裁;已接收自身 SLA+W,已返回 ACK	无 S1DAT 动作或	X	0	0	0	将接收数据字节并返回非 ACK 位
		无 S1DAT 动作	X	0	0	1	将接收数据字节并返回 ACK 位
70H	已接收通用调用地址(00H);已返回 ACK	无 S1DAT 动作或	X	0	0	0	将接收数据字节并返回非 ACK 位
		无 S1DAT 动作	X	0	0	1	将接收数据字节并返回 ACK 位
78H	主控器时在 SLA+R/W 中丢失仲裁; 已接收通用调用地址;已返回 ACK	无 S1DAT 动作或	X	0	0	0	将接收数据字节并返回非 ACK 位
		无 S1DAT 动作	X	0	0	1	将接收数据字节并返回 ACK 位
80H	前一次寻址使用自身从地址;已接收数据字节;已返回 ACK	读数据字节或	X	0	0	0	将接收数据字节并返回非 ACK 位
		读数据字节	X	0	0	1	将接收数据字节并返回 ACK 位
88H	前一次寻址使用自身从地址;已接收数据字节;已返回非 ACK	读数据字节或	0	0	0	0	切换到非寻址 SLV 模式;不识别自身 SLA 或通用调用地址
		读数据字节或	0	0	0	1	切换到非寻址 SLV 模式;识别自身 SLA;如果 S1ADR.0=1,将识别通用调用地址
		读数据字节或	1	0	0	0	切换到非寻址 SLV 模式;不识别自身 SLA 或通用调用地址;当总线空闲后发送起始条件
		读数据字节	1	0	0	1	切换到非寻址 SLV 模式;识别自身 SLA;如果 S1ADR.0=1,将识别通用调用地址;当总线空闲后发送起始条件
90H	前一次寻址使用通用调用;已接收数据字节;已返回 ACK	读数据字节或	X	0	0	0	将接收数据字节并返回非 ACK 位
		读数据字节	X	0	0	1	将接收数据字节并返回 ACK 位
98H	前一次寻址使用通用调用;已接收数据字节;已返回非 ACK	读数据字节或	0	0	0	0	切换到非寻址 SLV 模式;不识别自身 SLA 或通用调用地址
		读数据字节或	0	0	0	1	切换到非寻址 SLV 模式;识别自身 SLA;如果 S1ADR.0=1,将识别通用调用地址
		读数据字节或	1	0	0	0	切换到非寻址 SLV 模式;不识别自身 SLA 或通用调用地址;当总线空闲后发送起始条件
		读数据字节	1	0	0	1	切换到非寻址 SLV 模式;识别自身 SLA;如果 S1ADR.0=1,将识别通用调用地址;当总线空闲后发送起始条件
A0H	当使用 SLV/REC 或 SLV/TRX 静态寻址时,接收到停止条件或重复的起始条件	无 S1DAT 动作或	0	0	0	0	切换到非寻址 SLV 模式;不识别自身 SLA 或通用调用地址
		无 S1DAT 动作或	0	0	0	1	切换到非寻址 SLV 模式;识别自身 SLA;如果 S1ADR.0=1,将识别通用调用地址
		无 S1DAT 动作或	1	0	0	0	切换到非寻址 SLV 模式;不识别自身 SLA 或通用调用地址;当总线空闲后发送起始条件
		无 S1DAT 动作	1	0	0	1	切换到非寻址 SLV 模式;识别自身 SLA;如果 S1ADR.0=1,将识别通用调用地址;当总线空闲后发送起始条件

表 64 从发送器模式

状态码(S1STA)	I^2C 总线及 SIO1 硬件状态	应用软件的响应					SIO1 硬件执行的下一个动作
		读/写 S1DAT	写 S1CON				
			STA	STO	SI	AA	
A8H	已接收自身 SLA+R；已返回 ACK	装入数据字节或	X	0	0	0	将发送最后的数据字节并接收 ACK 位
		装入数据字节	X	0	0	1	将发送数据字节并接收 ACK 位
B0H	主控器时在 SLA+R/W 中丢失仲裁；已接收自身 SLA+R，已返回 ACK	装入数据字节或	X	0	0	0	将发送最后的数据字节并接收 ACK 位
		装入数据字节	X	0	0	1	将发送数据字节并接收 ACK 位
B8H	已发送 S1DAT 中数据字节；已返回 ACK	装入数据字节或	X	0	0	0	将发送最后的数据字节并接收 ACK 位
		装入数据字节	X	0	0	1	将发送数据字节并接收 ACK 位
C0H	已发送 S1DAT 中数据字节；已返回非 ACK	无 S1DAT 动作或	0	0	0	0	切换到非寻址 SLV 模式；不识别自身 SLA 或通用调用地址
		无 S1DAT 动作或	0	0	0	1	切换到非寻址 SLV 模式；识别自身 SLA；如果 S1ADR.0=1，将识别通用调用地址
		无 S1DAT 动作或	1	0	0	0	切换到非寻址 SLV 模式；不识别自身 SLA 或通用调用地址；当总线空闲后发送起始条件
		无 S1DAT 动作	1	0	0	1	切换到非寻址 SLV 模式；识别自身 SLA；如果 S1ADR.0=1，将识别通用调用地址；当总线空闲后发送起始条件
C8H	已发送 S1DAT 中最后的数据字节（AA=0）；已返回 ACK	无 S1DAT 动作或	0	0	0	0	切换到非寻址 SLV 模式；不识别自身 SLA 或通用调用地址
		无 S1DAT 动作或	0	0	0	1	切换到非寻址 SLV 模式；识别自身 SLA；如果 S1ADR.0=1，将识别通用调用地址
		无 S1DAT 动作或	1	0	0	0	切换到非寻址 SLV 模式；不识别自身 SLA 或通用调用地址；当总线空闲后发送起始条件
		无 S1DAT 动作	1	0	0	1	切换到非寻址 SLV 模式；识别自身 SLA；如果 S1ADR.0=1，将识别通用调用地址；当总线空闲后发送起始条件
F8H	无可用相关信息；SI=0	无 S1DAT 动作	无 S1DAT 动作				等待或进行当前的传输
00H	在 MST 或选择的从模式中，由于非法的起始或停止条件，使总线发生错误。当干扰导致 SIO1 进入一个未定义的状态时，也可产生状态 00H	无 S1DAT 动作	0	1	0	X	在 MST 或寻址 SLV 模式中只有内部硬件受影响。在所有情况下，总线被释放，而 SIO1 切换到非寻址 SLV 模式。STO 复位。

15.2.14.4 从发送器模式

在从发送器模式中，向主接收器发送一些数据字节（见图 40）。数据传输的初始化与从接收器模式相同。当 S1ADR 和 S1CON 初始化之后，SIO1 等待直到它被自身的从地址加上数据方向位（必须为 1）寻址。这样使 SIO1 操作在从发送器模式中。在接收到自身从地址和 R 位后，串行中断标志（SI）置位并可从 S1STA 中读出一个有效的状态代码。该状态代码用于指向一个中断服务程序，每个代码适当的执行动作见表 64。如果仲裁丢失但 SIO1 处于主模式，也可以进入从接收器模式（见状态 B0H）。

如果在传输中 AA 位复位，SIO1 将发送最后的数据字节并进入状态 C0H 或 C8H。如果 SIO1 继续进行传输，它将切换到非寻址从模式并忽略主接收器。这样主接收器将所有接收到的 1 当作串行数据。当 AA 复位后，SIO1 不响应自身的从地址或通用调用地址。但 I^2C 总线仍然被监控，而地址识别可随时通过置位

AA 而恢复。这意味着 AA 位可用于暂时将 SIO1 与 I^2C 总线隔离。

15.2.14.5 混合状态

有两个 S1STA 代码不对应于已定义的 SIO1 硬件状态（见表 65）。分别描述如下：

S1STA=F8H：

该状态代码指示由于串行中断标志 SI 还没有置位，所以无可用的相关信息。当 SIO1 没有参与串行传输时，该状态在其它状态之间产生。

S1STA=00H：

该状态代码指示在一个 SIO1 串行传输中已发生总线错误。当一个起始条件或停止条件出现在格式帧的非法位置时产生总线错误。这样的非法位置包括地址字节、数据字节或应答位的传输当中。当外部干涉扰乱了内部 SIO1 的信号时也会产生总线错误。当总线发生错误时，SI 置位。为了从总线错误中恢复，STO 标志必须置位而 SI 必须清零。这将使 SIO1 进入一个非寻址的从模式（一个已定义的状态）并清除 STO 标志（S1CON 中其它位不受影响）。SDA 和 SCL 线被释放（未发送停止条件）。

15.2.15 一些特殊情况

SIO1 具有处理下面在串行传输中可能出现的特殊情况的能力。

从两个主控器同时发出的重复起始条件：在主发送器或主接收器模式中可产生重复起始条件。如果另外一个主控器同时产生一个重复起始条件（见图 41），就出现一个特殊情况。在这发生之前，不会由主控器引起仲裁丢失，因为它们都发送相同的数据。如果 SIO1 硬件在它自身产生重复起始条件之前检测到 I^2C 总线上的重复起始条件，它将释放总线，不产生中断请求。如果另一个主控器通过产生停止条件释放总线，SIO1 将发送一个正常的起始条件（08H），并开始重试所有的数据传输。

15.2.15.2 对 I^2C 总线的强制访问

在一些应用中，一个非受控源可能引起总线挂起。这种情况下，可能会由干扰、总线的暂时中断或 SDA 和 SCL 之间暂时短路而引起麻烦。

如果一个非受控源产生一个多余的起始条件或屏蔽一个停止条件，那么 I^2C 总线就一直保持忙。如果 STA 标志置位并且不能在一定的时间内获得对总线的访问，就强制对 I^2C 总线进行访问。这通过在 STA 标志仍置位时将 STO 标志置位来实现。不发送停止条件。SIO1 硬件表现为好像已接收到停止条件并能够发送一个起始条件。STO 标志由硬件清零（见图 42）。

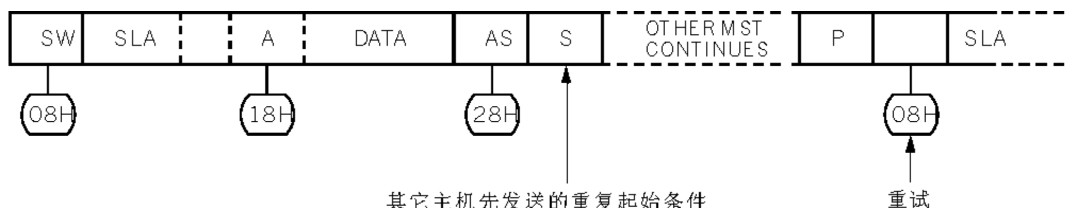


图 41 两个主控器同时产生的重复起始条件

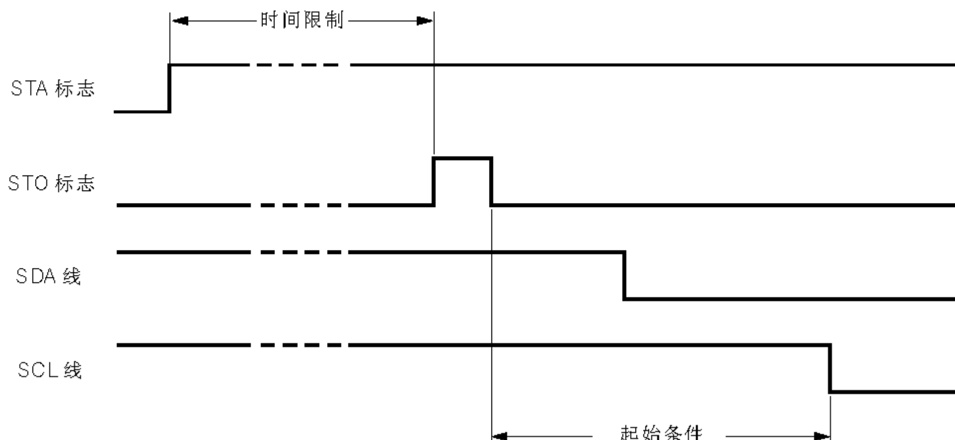


图 42 对一个被占用的 I^2C 总线的强制访问

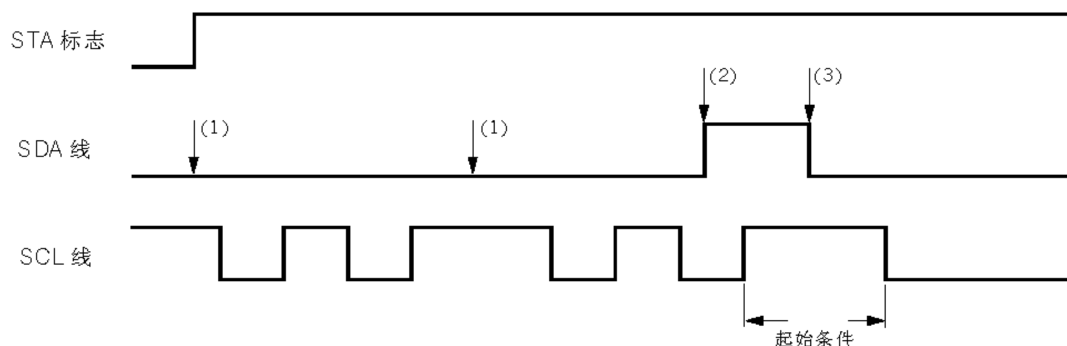
15.2.15.3 I²C 总线被 SCL 和 SDA 上的低电平阻塞

如果 SDA 或 SCL 被一个非受控源拉低，I²C 总线将被挂起。如果 SCL 线被总线上的一个器件阻塞（拉低），将无法进行串行传输。SIO1 的硬件将无法解决这类问题。发生这种情况时，必须由将 SCL 总线拉低的那个器件解决此问题。

如果 SDA 被总线上另一个器件所阻塞（例如，一个失去位同步的从器件），该问题可通过在 SCL 上发送额外的时钟脉冲来解决（见图 43）。SIO1 硬件当 STA 置位时发送额外的时钟脉冲。但由于 SDA 被拉低而 I²C 总线被认为是空闲的，因此不发送起始条件。

每经过两个额外的 SCL 时钟脉冲，SIO1 就试图产生一个起始条件。当 SDA 最终被释放时，发送一个正常的起始条件，进入状态 08H 并继续进行串行传输。

如果当 SDA 被阻塞（拉低）时产生对总线强制访问或发送重复起始条件，SIO1 像上面一样执行相同的动作。每种情况下，在成功发送起始条件后都进入状态 08H 并继续进行串行传输。注意，CPU 并未参与解决这些总线挂起的问题。



(1) 试图发送起始条件未成功
(2) SDA线释放
(3) 成功发送起始条件；进入状态D8H

图 43 从 SDA 低电平所引起的总线阻塞中恢复

15.2.15.4 总线错误

当起始或停止条件位于格式帧的非法位置时产生总线错误。例如，地址字节、数据字节或应答位的传输当中。

当作为主控器或从器件进行传输传输时，SIO1 硬件只对总线错误作出反应。当检测到总线错误时，SIO1 立即切换到非寻址的从模式，释放 SDA 和 SCL 并置位中断标志，将 00H 装入状态寄存器。该状态代码用于指向一个服务程序，该程序试图再次执行被中止的串行传输或只是简单的从错误状态中恢复，见表 65。

15.3 SI01 服务程序的软件示例

该章包括了软件示例，用于：

- 复位后对 SIO1 的初始化
- 进入 SIO1 中断服务程序
- 26 个状态服务程序用于
  - －主发送器模式
  - －主接收器模式
  - －从接收器模式
  - －从发送器模式

15.3.1 初始化

在初始化子程序中，SIO1 使能主和从两种模式。对于每种模式，分配一些内部数据 RAM 给 SIO1 作为发送或接收缓冲区。在该例中，保留 8 个字节内部数据 RAM 作为不同的用途。数据存储器的分布如图 44 所示。初始化子程序执行下列功能：

- SIADR 装入自身从地址和通用调用位（GC）
- P1.6 和 P1.7 位锁存装入逻辑 1
- RAM 位置 RADD 装入服务程序地址高字节
- 使能 SIO1 中断并置位中断优先级位
- 通过同时置位 S1CON 中的 ENS1 和 AA 位使能从模式，通过 S1CON 中的 CR0 和 CR1 定义串行时钟频率。主控器程序必须在主程序中启动。

SIO1 硬件开始检测 I^2C 总线用于自身的从地址和通用调用。如果通用调用或从地址被检测到，产生一个中断请求并向 S1STA 装入适当的状态信息。下面讲述一个快速进入适当服务程序的方法。

15.3.2 SIO1 中断服务程序

当进入 SIO1 中断时，首先将 PSW 压入堆栈，然后将 S1STA 和 HADD（由初始化程序装入的 26 个服务程序的高位地址字节）压入堆栈。S1STA 装载的代码为 26 个服务程序的低地址字节。下一条指令是 RET，从子程序中返回，高和低地址字节从堆栈中弹出不能给装入程序计数器。

下一条要执行的指令是状态服务程序的第一条指令。7 个字节的程序代码（在 8 个机器周期内执行）用于指向 26 个状态服务程序其中之一。

```
SI    PUSH   PSW        保存 PSW
      PUSH   S1STA      状态代码压栈（低地址字节）
      PUSH   HADD       高地址字节压栈
      RET               跳转到状态服务程序
```

状态服务程序位于程序存储器中的一个 256 字节页中。该页的位置由初始化程序定义。通过将数据 RAM 寄存器 HADD 装入页号，可将页定位在程序存储器中的任何位置。该例中所选的是页 01，服务程序位于地址 0100H 和 01FFH 之间。

15.3.3 状态服务程序

状态服务程序按照每隔 8 个地址位置分配。8 个字节代码对大多数服务程序来说是足够的。少数程序需要多于 8 个字节，必须跳转到其它位置以获得更多的代码字节。每个状态服务程序都是中断服务程序的一部分并处理 26 个状态其中之一。它以 RETI 指令结束，这样就返回到主程序。

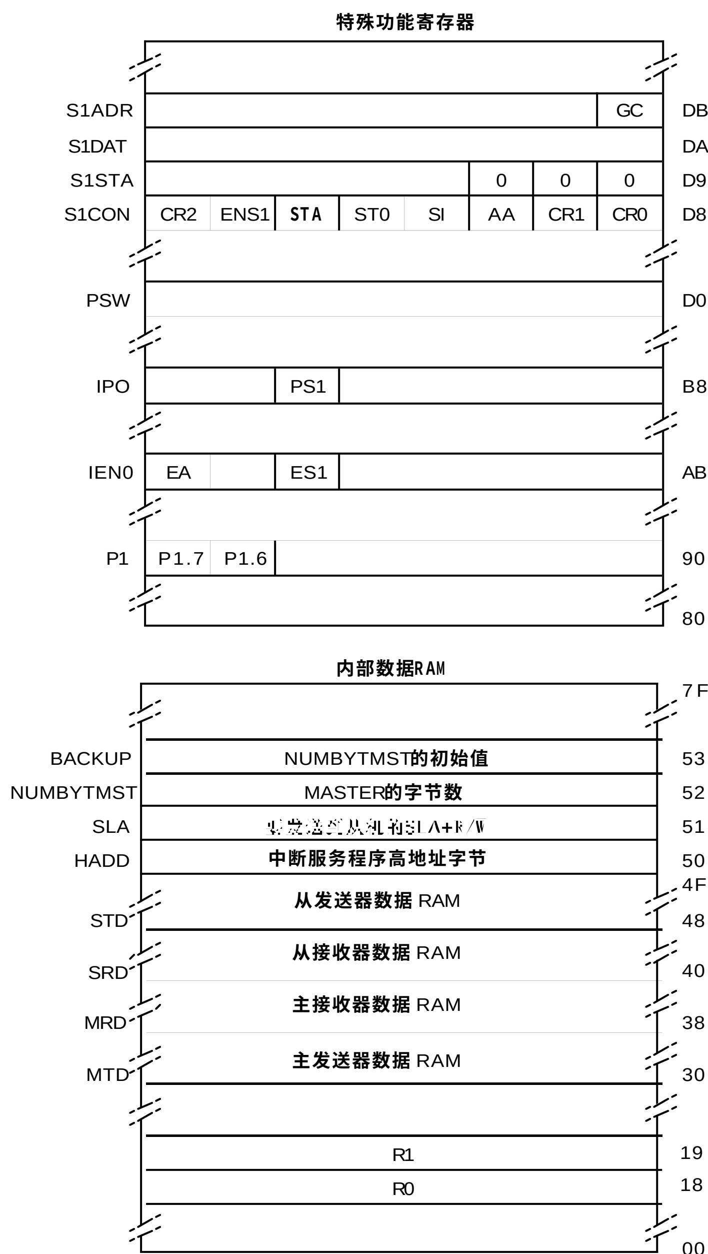


图 44 SIO1 数据存储分配

15.3.4 主发送器和主接收器模式

主模式从主程序中进入。要进入主发送器模式，主程序必须首先将从地址、数据字节和要发送的数据字节数装入内部数据 RAM。要进入主接收器模式，主程序必须首先装入从地址和要接收的数据字节数。R/W 位决定 SIO1 工作在主发送器还是主接收器模式。

当 S1CON 中的位 STA 通过 SETB 指令置位时，开始执行主模式操作。数据的传输按照表 61、表 62、图 37 和图 38 所示，由主状态服务程序控制在下面的例子当中，传输 4 个字节，没有重复起始条件。在丢失仲裁事件中，当总线变为空闲时重新启动传输。如果发生总线错误，I^2C 总线被释放而 SIO1 进入未选择的从接收器模式。如果从器件返回非应答信号，产生一个停止条件。

在状态代码 28H 和 58H 所指向的状态服务程序中，如果是 STA 标志而不是 STO 标志置位，串行传输中可包含一个重复的起始条件。必须增加额外的程序以决定在重复起始条件后所传输的数据。

15.3.5 从发送器和从接收器模式

在初始化之后，SIO1 继续测试 I^2C 总线并当它检测到自身的从地址或通用调用地址时（见表 63、表 64、图 39 和图 40），进入其中一个从状态服务程序。如果在主模式中丢失仲裁，主模式在当前传输之后重

新启动。如果发生总线错误，I^2C 总线被释放，SIO1 进入未选择的从接收器模式。

在从接收器模式中，最多 8 个接收数据可保存在内部数据 RAM 中。8 字节最大值确保了在主器件发送更多字节时，其它 RAM 位置不会被覆盖如果发送的数据超过 8 个字节，不会返回应答位，并且 SIO1 进入非寻址从接收器模式。在检测到通用调用地址后，最多只能将一个接收字节保存到内部数据 RAM。如果发送的字节超过 1 个，返回非应答位并且 SIO1 进入非寻址的从接收器模式。

在从发送器模式中，数据可从之前由主程序装入的内部数据 RAM 的同一位置获得。在主接收器件返回一个非应答位后，SIO1 进入非寻址从模式。

15.3.6 使软件适合不同的应用

下面的软件示例展示了中断服务程序的典型结构，其中包括 26 个状态服务程序。可作为用户应用的基础。如果 4 个模式中的一个或多个没有被使用，相关的状态服务程序可以移去，但要注意的是，删除的程序不能被调用。

该示例不包括任何超时程序。在从模式中，超时程序并不是很有用，因为这些模式中，SIO1 本质上是一个无源器件。在主模式中，如果串行传输没有在规定的时间内完成，可使用内部定时器产生超时。该时间周期由连接到 I^2C 总线的系统决定。

软件清单

```
;************************************************************************************
; SIO1特殊功能寄存器位置
;************************************************************************************
S1CON           -0xd8
S1STA           -0xd9
S1DAT           -0xda
S1ADR           -0xdb
IEN0            -0xa8
IP0             02b8
;************************************************************************************
; 位位置
;************************************************************************************
STA             -0xdd                           ; S1CON中的STA位
SIO1HP          -0xbd                           ; IP0, SIO1优先级位
;************************************************************************************
; 写入S1CON的立即数
;************************************************************************************
ENS1_NOTSTA_STO_NOTSI_AA_CR0            -0xd5   ; Generates STOP
                                                ; (CR0 = 100kHz @ f OSC =6MHz)
ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0         -0xc5   ;释放总线和应答
;
ENS1_NOTSTA_NOTSTO_NOTSI_NOTAA_CR0      -0xc1   ;释放总线和非应答
;
ENS1_STA_NOTSTO_NOTSI_AA_CR0            -0xe5   ;释放总线和置位STA
;************************************************************************************
; 通用立即数
;************************************************************************************
OWNSLA          -0x31                           ;自身SLA+通用调用位必须写入S1ADR
ENSIO1          -0xa0                           ; EA+ES1,SIO1中断使能必须写入IEN0
```

```
PAG1        -0x01                     ; 选择 PAG1 作为 HADD
SLAW        -0xc0                     ; 要发送的SLA+W
SLAR        -0xc1                     ; 要发送的SLA+R
SELRB3      -0x18                     ; 选择寄存器 Bank 3
;*************************************************************************************
; 数据RAM中的位置
;*************************************************************************************
MTD         -0x30                     ; MST/TRX/DATA 基本地址
MRD         -0x38                     ; MST/REC/DATA基本地址
SRD         -0x40                     ; SLV/REC/DATA基本地址
STD         -0x48                     ; SLV/TRX/DATA基本地址
BACKUP      -0x53                     ; NUMBYTMST的备份
                                      ; 为了在仲裁丢失时恢复NUMBYTMST
NUMBYTMST  -0x52                      ; 作为MST时要发送或接收的字节数
SLA         -0x51                     ; 包含了要发送的SLA+R/W
HADD        -0x50                     ; 用于状态0到状态25的高地址字节
;*************************************************************************************
; 初始化程序
; IIC接口作为从接收器或从发送器的初始化并启动主机发送或主机接收功能。要发送或接收的是4个字节
;*************************************************************************************
.sect     strt
.base     0x00
            ajmp      INIT                ; RESET
.sect     initial
.base     0x200
INIT:       mov       S1ADR,#OWNSLA       ; 装载自身SLA + 使能通用调用识别
            setb      P1(6)               ; P1.6 高电平
            setb      P1(7)               ; P1.7 高电平
            mov       HADD,#PAG1
            orl       IEN0,#ENSIO1        ; 使能SIO1中断
            clr       SIO1HP              ; SIO1中断低优先级
            mov       S1CON, #ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                          ; 初始化SLV 功能
;*************************************************************************************
; 启动主机发送功能
;*************************************************************************************
            mov       NUMBYTMST,#0x4      ; 发送4个字节
            mov       SLA,#SLAW           ; SLA+W, 发送功能
            setb      STA                 ; 置位S1CON中的STA
;*************************************************************************************
; START MASTER RECEIVE FUNCTION启动主机接收功能
;*************************************************************************************
            mov       NUMBYTMST,#0x4      ; 接收4个字节
            mov       SLA,#SLAR           ; SLA+R,接收功能
```

```
                setb      STA               ; 置位 S1CON 中的 STA
;*****************************************************************************************
; SIO1 INTERRUPT ROUTINE中断服务程序
;*****************************************************************************************
.sect     intvec                            ; SIO1中断向量
.base     0x00
; S1STA 和 HADD are pushed onto the stack.压入堆栈
; 它们作为RET指令的返回地址
; RET指令使PC指向地址HADD和S1STA，并跳转到正确的子程序
                push      psw               ; 保存psw
                push      S1STA
                push      HADD
                ret                         ; 跳转到地址HADD,S1STA.
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 状态： 00, 总线错误
; 动作：.进入非寻址SLV模式并释放总线，STO复位
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -.
sect      st0
.base     0x100
                mov       S1CON,#ENS1_NOTSTA_STO_NOTSI_AA_CR0 ; clr SI
                                            ; 置位 STO,AA
                pop       psw
                reti
;*****************************************************************************************
; 主机状态服务程序
;*****************************************************************************************
;*****************************************************************************************
; 状态 08 和状态10 都用于 MST/TRX and MST/REC.
; The R/W 位决定下个状态是在MST/TRX 模式中还是在MST/REC模式中
;*****************************************************************************************
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 状态 : 08, A, 起始条件已发送
; 动作 : SLA+R/W 已发送，已接收ACK位
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -.
sect      mts8
.base     0x108
                mov       S1DAT,SLA         ; 装载SLA+R/W
                mov       S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                            ; 清零SI
                ajmp      INITBASE1
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 状态 : 10, 已发送一个重复的起始条件
; 动作 : SLA+R/W 已发送, 已接收ACK位
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -.
```

```
sect        mts10
.base       0x110
            mov     S1DAT,SLA           ; 装载 SLA+R/W
            mov     S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                        ; 清零 SI
            ajmp    INITBASE1
.sect       ibase1
.base       0xa0
INITBASE1:  mov     psw,#SELRB3
            mov     r1,#MTD
            mov     r0,#MRD
            mov     BACKUP,NUMBYTMST ;保存初始化值
            pop     psw
            reti
;*************************************************************************************
; 主发送器状态服务程序
;*************************************************************************************
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 状态 : 18, 前一状态为8 或10, SLA+W 已发送，已接收ACK位
; 动作 : 已发送第一个数据, 已接收ACK位
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -dc
.sect       mts18
.base       0x118
            mov     psw,#SELRB3
            mov     S1DAT,@r1
            ajmp    CON
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 状态 : 20, SLA+W 已发送, 已接收非ACK位
; 动作 : 发送停止条件
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -.
sect        mts20
.base       0x120
            mov     S1CON,#ENS1_NOTSTA_STO_NOTSI_AA_CR0
                                        ; 置位STO, 清零 SI
            pop     psw
            reti
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 状态 : 28, S1DAT的数据已发送，已接收ACK
; 动作 : 如果发送的数据是最后一个，那么发送一个停止条件，否则发送下一个数据
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -.
sect        mts28
.base       0x128
            djnz    NUMBYTMST,NOTLDAT1 ; 如果不是最后一个数据就跳转
            mov     S1CON,#ENS1_NOTSTA_STO_NOTSI_AA_CR0
```

```
                                        ; 清零SI, 置位AA
          ajmp      RETmt
.sect     mts28sb
.base     0x0b0
NOTLDAT1: mov       psw,#SELRB3
          mov       S1DAT,@r1
CON:      mov       S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                        ; 清零 SI, 置位 AA
          inc       r1
RETmt :   pop       psw
          reti
;-------------------------------------------------------------
; 状态 : 30, DATA of S1DAT 已发送, NOT ACK已接收
; 动作 : 发送一个停止条件
;-------------------------------------------------------------.
sect      mts30
.base     0x130
          mov       S1CON,#ENS1_NOTSTA_STO_NOTSI_AA_CR0
                                        ; 置位 STO, 清零 SI
          pop       psw
          reti;
;-------------------------------------------------------------
; 状态 : 38, 仲裁丢失在SLA+W 或 DATA.
; 动作 : 总线被释放，进入非寻址SLV 模式. 当IIC 总线被再次释放时，发送一个新的起始条件。
;-------------------------------------------------------------
.sect     mts38
.base     0x138
          mov       S1CON,#ENS1_STA_NOTSTO_NOTSI_AA_CR0
          mov       NUMBYTMST,BACKUP
          ajmp      RETmt
;*****************************************************************************************
; 主接收器状态服务程序
;*****************************************************************************************
;-------------------------------------------------------------
; 状态 : 40, 前一状态为08 或10, SLA+R 已发送, ACK 已接收
; 动作 : 数据将被接收, 返回ACK
;-------------------------------------------------------------.
sect      mts40
.base     0x140
          mov       S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                        ;清零 STA, STO, SI 置位 AA
          pop       psw
          reti
;-------------------------------------------------------------
```

```
; 状态 : 48, SLA+R已发送, 非ACK 已接收
; 动作 : 将产生停止条件
;-----------------------------------------------------------------.
sect        mts48
.base       0x148
STOP:           mov         S1CON,#ENS1_NOTSTA_STO_NOTSI_AA_CR0
                                                  ; 置位 STO, 清零 SI
                pop         psw
                reti
;-----------------------------------------------------------------
; 状态 : 50, 数据已接收, ACK 返回
; 动作 : 读S1DAT数据. DATA 将被接收,如果是最后的数据则返回非ACK, 否则返回ACK
;-----------------------------------------------------------------.
sect        mrs50
.base       0x150
                mov         psw,#SELRB3
                mov         @r0,S1DAT ; 读已接收的数据
                ajmp        REC1
.sect       mrs50s
.base       0xc0
REC1:           djnz        NUMBYTMST,NOTLDAT2
                mov         S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_NOTAA_CR0
                                                  ; 清零SI,AA
                sjmp        RETmr
NOTLDAT2:       mov         S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                  ; 清零 SI, 置位AA
RETmr:          inc         r0
                pop         psw
                reti
;-----------------------------------------------------------------
; 状态 : 58, 数据已接收, 非ACK 返回
; 动作 : 读主状态服务程序的数据并产生一个停止条件
;-----------------------------------------------------------------.
sect        mrs58
.base       0x158
                mov         psw,#SELRB3
                mov         @R0,S1DAT
                sjmp        STOP
;*****************************************************************************************
; 从接收器状态服务程序
;*****************************************************************************************
;-----------------------------------------------------------------
; 状态 : 60, 自身SLA+W 已接收, ACK 返回
; 动作 : 数据将被接收并返回 ACK
```

```
;-------------------------------------------------------------.
sect      srs60
.base     0x160
          mov     S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                          ; 清零SI, 置位 AA
          mov     psw,#SELRB3
          ajmp    INITSRD
.sect     insrd
.base     0xd0
INITSRD:  mov     r0,#SRD
          mov     r1,#8
          pop     psw
          reti
;-------------------------------------------------------------
; 状态 : 68, 仲裁丢失在SLA 和 R/W 作为 MST 自身SLA+W 已接收, ACK 返回
; 动作 : 数据将被接收并返回 ACK. 在总线再次空闲后, 置位STA以重新启动 MST模式
;-------------------------------------------------------------
sect      srs68
.base     0x168
          mov     S1CON,#ENS1_STA_NOTSTO_NOTSI_AA_CR0
          mov     psw,#SELRB3
          ajmp    INITSRD
;-------------------------------------------------------------
; 状态 : 70, General call已接收, ACK 返回
; 动作 : 数据将被接收并返回 ACK
;-------------------------------------------------------------.
sect      srs70
.base     0x170
          mov     S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                          ; 清零 SI, 置位 AA
          mov     psw,#SELRB3             ; 初始化 SRD 计数器
          ajmp    initsrd
;-------------------------------------------------------------
; 状态 : 78, Arbitration lost in SLA+R/W as MST. General call has been received, ACK returned.
; 动作 : 数据将被接收并返回ACK. 置位STA,在总线空闲之后以重新启动 MST模式
;-------------------------------------------------------------.
sect      srs78
.base     0x178
          mov     S1CON,#ENS1_STA_NOTSTO_NOTSI_AA_CR0
          mov     psw,#SELRB3             ; 初始化 SRD 计数器
          ajmp    INITSRD
;-------------------------------------------------------------
; 状态 : 80, 之前使用自身SLA寻址. 数据已接收并返回 ACK.
; 动作 : 读数据
```

```
；如果接收的数据是最后一个，那么剩余的数据将被接收并返回 非ACK,否则下一个数据将被接收并返回
ACK;
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -.
sect        srs80
.base       0x180
            mov         psw,#SELRB3
            mov         @r0,S1DAT                  ；读接收的数据
            ajmp        REC2
.sect       srs80s
.base       0xd8
REC2:       djnz        r1,NOTLDAT3
LDAT:       mov         S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_NOTAA_CR0
                                                   ；清零 SI,AA
            pop         psw
            reti
NOTLDAT3: mov           S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                   ；清零 SI，置位 AA
            inc         r0
RETsr:      pop         psw
            reti
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
；状态 ：88，之前使用自身SLA寻址．数据已接收并返回非ACK
；动作： 无数据保存，进入非寻址SLV模式．识别自身SLA．如果S1ADR. 01，识别General call.
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
.sect       srs88
.base       0x188
                mov         S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                   ；清零 SI，置位 AA
                ajmp        RETsr
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
；状态 ：90，之前使用general call寻址．数据已接收并返回非ACK.
；动作： 读数据.
；在General call之后只有一个数据和ACK将被接收，第二个数据将被接收并返回非ACK.
；数据将被接收并返回非 ACK.
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
.sect       srs90
.base       0x190
                mov         psw,#SELRB3
                mov         @r0,S1DAT              ；读接收到的数据
                ajmp        LDAT
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
；状态 ：98，之前使用general call寻址.
；数据将被接收并返回非 ACK
；动作： 无数据保存，进入非寻址SLV模式
```

```
; 识别自身SLA. 如果S1ADR. 01,识别General call;
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
.sect       srs98
.base       0x198
            mov         S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                    ; 清零 SI, 置位 AA
            pop         psw
            reti
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 状态 : A0, 当仍然作为SLV/REC 或 SLV/TRX时, 接收一个停止条件或重复起始条件
; 动作 : 无数据保存, 进入非寻址SLV模式
; 识别自身SLA. 如果S1ADR. 01,识别General call.
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -.
sect        srsA0
.base       0x1a0
            mov         S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                    ; 清零 SI, 置位 AA
            pop         psw
            reti
;*****************************************************************************************
; 从发送器服务程序
;*****************************************************************************************
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 状态 : A8, 接收自身SLA+R, 返回ACK
; 动作 : 数据将被接收并返回 ACK
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
.sect       stsa8
.base       0x1a8
            mov         S1DAT,STD           ;装载S1DAT中的数据
            mov         S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                    ; 清零 SI, 置位 AA
            ajmp        INITBASE2
.sect       ibase2
.base       0xe8
INITBASE2:  mov         psw,#SELRB3
            mov         r1, #STD
            inc         r1
            pop         psw
            reti
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; 状态 : B0, 当作为MST时, 在SLA和 R/W中丢失仲裁. 接收自身SLA+R, 返回ACK.
; 动作 : 数据将被接收并返回 ACK.
; 置位STA,在总线空闲之后以重新启动 MST模式.
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -.
```

```
sect        sstsb0
.base       0x1b0
            mov     S1DAT,STD               ; 装载 S1DAT中的数据
            mov     S1CON,#ENS1_STA_NOTSTO_NOTSI_AA_CR0
            ajmp    INITBASE2
;-----------------------------------------------------------------
; 状态 : B8, 数据已发送, ACK 已接收
; 动作 : 数据将被发送并接收ACK
;-----------------------------------------------------------------
.sect       stsb8
.base       0x1b8
            mov     psw,#SELRB3
            mov     S1DAT,@r1
            ajmp    SCON
.sect       scn
.base       0xf8
SCON:       mov     S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                            ; 清零 SI, 置位 AA
            inc     r1
            pop     psw
            reti
;-----------------------------------------------------------------
; 状态: C0, 数据已发送并接收非ACK
; 动作 : 进入非寻址SLV模式.
;-----------------------------------------------------------------
.sect       stsc0
.base       0x1c0
            mov     S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                            ; 清零 SI, 置位 AA
            pop     psw
            reti
;-----------------------------------------------------------------
; 状态 : C8, 最后一个数据已发送 (AA=0), 接收到ACK.
; 动作 : 进入非寻址SLV模式
;-----------------------------------------------------------------
.sect       stsc8
.base       0x1c8
            mov     S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                            ; 清零 SI, 置位 AA
            pop     psw
            reti
;***************************************************************************************
; SIO1中断服务程序结束
;***************************************************************************************
```

16 定时器 2 的操作

16.1 定时器 2

定时器 2 是一个 16 位定时/计数器，其中包括两个寄存器 TMH2（高字节）和 TML2（低字节）。它可被关闭或通过一个预分频器由下列两个时钟源之一驱动：fclk/6 或一个外部信号。当定时器 T2 配置为计数器时，预分频器由 T2（P3.0）脚上的外部信号提供时钟。T2 的上升沿使预分频器加一，最大的重复率为每机器周期一个计数（6MHz 振荡器时为 1MHz）。

定时器 T2 的最大重复率是定时器 0 和 1 的两倍。T2（P3.0）在每个机器周期的 S2P1 和 S5P1 采样（即每个机器周期两次采样）。当 T2 在一次采样时为低而下一次采样时为高，则检测到一个上升沿。为了确保检测到上升沿，输入信号必须至少保持 1/2 周期低电平然后至少保持 1/2 周期高电平。如果在 S2P1 结束前检测到上升沿，定时器将在下个周期加一。否则将在一个周期之后增加。预分频器具有可编程的分频系数 1，2，4 或 8，如果分频系数或输入源改变或定时/计数器复位，则清零。

定时器 T2 可在运行中读出，但不具有额外的读锁存。必须执行软件上的预防措施以避免在 T2 被读时低到高字节溢出所造成的错误判断。定时器 T2 不可装载，由 RST 信号或输入信号 RT2（如果使能）的上升沿复位。RT2 通过置位 T2ER(TM2CON.5)使能。

当定时器的低字节或 16 位溢出时，可能会产生一个中断请求。溢出可编程以请求一个中断。这两种情况下，中断向量是相同的。当低字节溢出（TML2）时，标志 T2B0(TM2CON)置位；而当 TMH2 溢出时，标志 T20V(TM2IR)置位。这些标志在溢出发生的一个周期后置位。注意当 T20V 置位时 T2B0 也置位。要使能字节溢出中断，位 ET2（IE1.7,使能溢出中断，见表 67）和 T2IS0（TM2CON.6,字节溢出中断选择）必须置位。位 TWBO（TM2CON.4）为定时器 T2 的字节溢出标志。要使能 16 位溢出中断，位 ET2（IE1.7,使能溢出中断）和 T2IS1（TM2CON.7,16 位溢出中断选择）必须置位。位 T2OV（TM2IR.7）为定时器 T2 的 16 位溢出标志。所有中断都必须由软件复位。要同时使能字节和 16 位溢出，T2IS0 和 T2IS1 必须置位并请求两个中断服务程序。对溢出标志的测试指示必须执行哪一个服务程序。对每个服务程序，只有相应的溢出标志位必须清零。如果定时器 T2 外部复位使能位（TM2CON 中的 T2ER）置位，定时器 T2 可通过 RT2（P3.1）上的上升沿复位。此复位也将清零预分频器。在空闲模式中，定时/计数器和预分频器复位并停止。定时器 T2 由 TM2CON 控制（见 16.1.1）。

表 66 定时器 T2 中断使能寄存器 IEN1（地址 E8H）

7	6	5	4	3	2	1	0
ET2	ECAN	ECM1	ECM0	ECT3	ECT2	ECT1	ECT0

表 67 中断使能寄存器 IEN1 位描述

位	符号	功能
7	ET2	使能定时器 T2 溢出中断
6	ECAN	使能 CAN 中断
5	ECM1	使能 T2 比较器 1 中断
4	ECM0	使能 T2 比较器 0 中断
3	ECT3	使能 T2 捕获寄存器 3 中断
2	ECT2	使能 T2 捕获寄存器 2 中断
1	ECT1	使能 T2 捕获寄存器 1 中断
0	ECT0	使能 T2 捕获寄存器 0 中断

16.1.1 T2 控制寄存器（TM2CON）

表 68 T2 控制寄存器（地址 EAH）

7	6	5	4	3	2	1	0
T2IS1	T2IS0	T2ER	T2BO	T2P1	T2P0	T2MS1	T2MS0

表 69 TM2CON 位描述

位	符号	功能
7	T2IS1	定时器 T2 16 位溢出中断选择
6	T2IS0	定时器 T2 字节溢出中断选择
5	T2ER	定时器 T2 外部复位使能。当该位置位时，定时器 T2 可通过 RT2（P3.0）上的上升沿复位。
4	T2BO	定时器 T2 字节溢出中断标志
3	T2P1	定时器 T2 预分频器选择（见表 70）
2	T2P0	
1	T2MS1	定时器 T2 模式选择（见表 71）
0	T2MS0	

表 70　定时器 2 预分频器选择

T2P1	T2P0	定时器 T2 时钟
0	0	时钟源
0	1	1/2 时钟源
1	0	1/4 时钟源
1	1	1/8 时钟源

表 71　定时器 2 模式选择

T2MS1	T2MS0	模式选择
0	0	定时器 T2 停止（关闭）
0	1	1/6 fclk T2 时钟源
1	0	测试模式；未用
1	1	定时器 T2 时钟源＝脚 T2

16.1.2　定时器 T2 扩展

当使用一个 6MHz 振荡器时，定时器 T2 每 65.5，131，262 或 524ms（这取决于预分频器的分频系数）产生一次 16 位溢出。即最大周期时间约为 0.5 秒。在周期时间大于 0.5 秒的应用中需要对 T2 进行扩展。可以这样来实现：选择 $f_{CLK}/6$ 作为时钟源（置位 T2MS0，清零 T2MS1），将预分频系数设定为 1/8（置位 T2P0 和 T2P1），禁止字节溢出中断（清零 T2IS0）并使能 16 位溢出中断（置位 T2IS1）。下面的软件示例使用了 3 个字节的扩展，最大周期时间大约为 2400 小时。

```
OVINT:  PUSH    ACO         ；保存ACC
        PUSH    PSW         ；保存PSW
        INC     TlMEX1      ；增加扩展定时器第一字节（低位）
        MOV     A,TlMEX1
        JNZ     INTEX       ；如果无溢出，跳转到INTEX
        INC     TlMEX2      ；增加第二个字节
        MOV     A,TlMEX2
        JNZ     INTEX       ；如果无溢出，跳转到INTEX
        INC     TlMEX3      ；增加第三个字节（高位）
INTEX:  CLR     T2OV        ；清除中断标志
        POP     PSW         ；恢复PSW
        POP     ACC         ；恢复ACC
        RETI                ；中断返回
```

16.1.3　定时器 2，捕获和比较逻辑

定时器 2 与 4 个 16 位捕获寄存器和 3 个 16 位比较寄存器相连接。捕获寄存器可在对应输入脚发生电平跳变时捕获定时器 2 的内容。比较寄存器可用于以一定的可编程的时间间隔设置或复位 P3 的输出管脚。

定时器 T2 和捕获以及比较逻辑的组合在包括旋转机械、自动注射系统等应用中具有强大的功能。定时器 2，捕获和比较逻辑如图 45 所示。

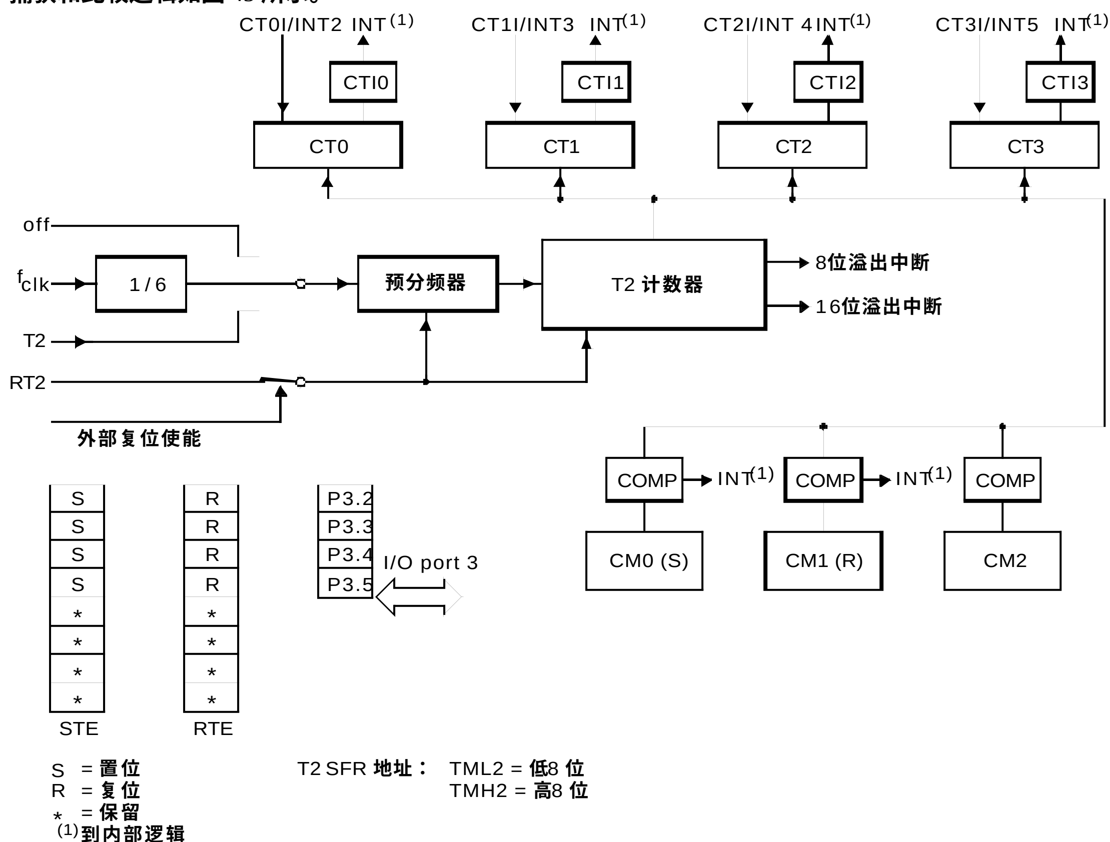


图 45 定时器 T2 方框图

16.1.4 捕获逻辑

与定时器 T2 连接的 4 个捕获寄存器分别是：CT0,CT1,CT2 和 CT3。这些寄存器装载定时器 T2 的内容并在接收到 CT0I,CT1I,CT2I 和 CT3I 时请求一个中断。4 个中断标志都位于定时器 T2 中断寄存器（TM2IR）。如果不要求捕获功能，这些输入可看作是附加的外部中断输入（INT2~INT5）。

使用捕获寄存器 CTCON（见 16.1.4.1）时，这些输入可在上升沿、下降沿或上升及下降沿进行捕获。输入信号在每个周期的 S1P1 被采样。当检测到所选择的边沿，T2 定时器的内容在周期结束时被捕获。

16.1.4.1 捕获控制寄存器（CTCON）

表 72 捕获控制寄存器（地址 EBH）

7	6	5	4	3	2	1	0
CTN3	CTP3	CTN2	CTP2	CTN1	CTP1	CTN0	CTP0

表 73 CTCON 位描述

位	符号	描述
7	CTN3	CT3I 的下降沿触发捕获寄存器 3
6	CTP3	CT3I 的上升沿触发捕获寄存器 3
5	CTN2	CT2I 的下降沿触发捕获寄存器 2
4	CTP2	CT2I 的上升沿触发捕获寄存器 2
3	CTN1	CT1I 的下降沿触发捕获寄存器 1
2	CTP1	CT1I 的上升沿触发捕获寄存器 1
1	CTN0	CT0I 的下降沿触发捕获寄存器 0
0	CTP0	CT0I 的上升沿触发捕获寄存器 0

16.1.5 使用寄存器测量时间间隔

当一个连续的外部事件以上升沿或下降沿的形式出现在捕获输入脚时，可使用定时器 T2 和捕获寄存器测量两个事件的间隔。当发生一个事件时，定时器 T2 的内容复制到相关的捕获寄存器中并产生一个中断请求。如果知道上次事件发生时 T2 的内容，中断服务程序就可计算时间间隔。使用 6MHz 振荡器时，定时器 T2 可编程为每 524ms 溢出一次。当事件的间隔小于该时间时，对时间间隔的计算就很简单，而且中断服务程序很短。对于更长的间隔时间，使用定时器 T2 扩展程序。

16.1.6 比较逻辑

每次定时器 T2 增加时，3 个 16 位比较寄存器 CM0,CM1 和 CM2 与定时器 T2 的新值进行比较。当发生匹配时，TM2IR 中对应的中断标志位在下个周期结束时置位。当 CM0 发生匹配时，控制器置位 P3 口的位 0～3（如果对应的置位使能寄存器 STE 位为逻辑 1，见 16.6.2）。

当 CM1 发生匹配时，控制器复位 P3 口的位 0～3（如果对应的复位/使能寄存器 RTE 的位为逻辑 1，见 16.6.1）。如果 RTE 为 0，当 CM1 或 CM2 与定时器 T2 发生匹配时 P3.n 不受影响。

如果当前操作作为“置位”，下一个操作将是“复位”，即使口锁存在“复位”操作出现之前由软件复位。CM0,CM1 和 CM2 由 RST 信号复位。

更改后的口锁存信息在发生匹配周期的下一个周期的 S5P1 输出到管脚。如果口的值由软件更改，输出将在下个周期的 S1P1 改变。P3 口位 0～3 在任何时候都可由软件置位或复位。比较器匹配所导致的硬件更改优先于同一个周期内软件的更改。当比较器结果同时要求一个“置位”和一个“复位”时，口锁存将复位。

16.1.6.1 复位/翻转使能寄存器（RTE）

表 74 复位/翻转使能寄存器（地址 EFH）

7	6	5	4	3	2	1	0
－	－	－	－	RP35	RP34	RP33	RP32

表 73 RTE 位描述

位	符号	描述
7~4	－	保留
3	RP35	如果为高，当 CM2 与 T2 发生匹配时 P3.5 复位
2	RP34	如果为高，当 CM2 与 T2 发生匹配时 P3.4 复位
1	RP33	如果为高，当 CM2 与 T2 发生匹配时 P3.3 复位
0	RP32	如果为高，当 CM2 与 T2 发生匹配时 P3.2 复位

16.1.6.2 置位使能寄存器（STE）

表 74 置位使能寄存器（地址 EEH）

7	6	5	4	3	2	1	0
－	－	－	－	SP35	SP34	SP33	SP32

表 73 STE 位描述

位	符号	描述
7~4	－	保留
3	SP35	如果为高，当 CM2 与 T2 发生匹配时 P3.5 置位
2	SP34	如果为高，当 CM2 与 T2 发生匹配时 P3.4 置位
1	SP33	如果为高，当 CM2 与 T2 发生匹配时 P3.3 置位
0	SP32	如果为高，当 CM2 与 T2 发生匹配时 P3.2 置位

16.1.7 定时器 T2 中断标志寄存器 TM2IR

8 个定时器 T2 中断标志中的 7 个位于特殊功能寄存器 TM2IR 中（见 16.1.7.1）。第 8 个标志为 TM2CON.4。

当定时器 T2 的内容被捕获时，CT0I 和 CT1I 标志在周期的 S4 置位。中断逻辑在 S2 时对 CT0I 进行扫描，在 S3 对 CT1I 进行扫描。CT2I 和 CT3I 在 S6 时置位并在 S4 和 S5 被扫描。相关的中断请求在下个周期识别。如果这些标志被轮询，由于寄存器在 S5 被读出，CT0I 或 CT1I 的跳变将在 CT2I 或 CT3I 的跳变前一个周期被识别，CMI0,CMI1 和 CMI2 标志在匹配之后的周期的 S6 置位。中断逻辑在 S2 对 CMI0 进行扫描；CMI1 和 CMI2 在 S3 和 S4 被扫描。CMI0 和 CMI1 的匹配在匹配发生的两个周期后由中断逻辑（或通过轮询标志）识别。CMI2 的匹配不会产生中断，该标志只能被轮询。

16 位溢出标志（T2OV）和字节溢出标志（T2BO）在溢出发生周期的 S6 置位。这些标志由中断逻辑在下个周期识别。特殊功能寄存器 IP1（见 16.1.7.2）用于决定定时器 T2 的优先级。置位该位将得到高优先级。清零该位将得到低优先级。

16.1.7.1 中断标志寄存器（TM2IR）

表 78 中断标志寄存器（地址 C8H）

7	6	5	4	3	2	1	0
T2OV	CMI2/CAN	CMI1	CMI0	CTI3	CTI2	CTI1	CTI0

表 73 TM2IR 位描述

位	符号	描述
7	T2OV	T2：16 位溢出标志
6	CMI2/CAN	CM2：标志（只可轮询） CAN：CAN 中断标志（只可轮询）
5	CMI1	CM1：中断标志
4	CMI0	CM0：中断标志
3	CTI3	CT3：中断标志
2	CTI2	CT2：中断标志
1	CTI1	CT1：中断标志
0	CTI0	CT0：中断标志

16.1.7.2 中断优先级寄存器 1（IP1）

表 80 中断优先级寄存器（地址 F8H）

7	6	5	4	3	2	1	0
PT2	PCAN	PCM1	PCM0	PCT3	PCT2	PCT1	PCT0

表 73 IP1 位描述

位	符号	描述
7	PT2	T2 溢出中断优先级
6	PCAN	CAN 中断优先级
5	PCM1	T2 比较器 1 中断优先级
4	PCM0	T2 比较器 0 中断优先级
3	PCT3	T2 捕获寄存器 3 中断优先级
2	PCT2	T2 捕获寄存器 2 中断优先级
1	PCT1	T2 捕获寄存器 1 中断优先级
0	PCT0	T2 捕获寄存器 0 中断优先级

17 看门狗定时器（T3）

除了定时器 T2 和标准定时器之外，PP8xC591 还集成了一个看门狗定时器 T3。看门狗的作用是在微控制器进入错误的处理器状态（可能由电噪声或 RFI 所导致）的一定时间内使其复位。当看门狗使能时，如果用户程序没有在一个规定的时间长度（看门狗间隔）内重装看门狗，看门狗电路将产生一个系统复位信号。

看门狗电路描述

看门狗定时器（定时器 T3）包括一个带有 11 位预分频器的 8 位定时器，见图 46。预分频器由振荡器频率的 1/6（6MHz 振荡器时为 1MHz）驱动。8 位定时器每“ t ”秒加一。T3 由 12MHz 振荡器频率驱动时，每 1024μs 加一。公式如下：

t = 6×2048×1/fclk = 1024μs　　此处 fclk = 12MHz

如果 8 位定时器溢出，将产生一个短的内部复位脉冲使 P8xC591 复位。此复位脉冲也输出到 RST 脚。如果 RST 脚连接了电容，此复位脉冲（3 个机器周期）将被破坏。但这不会影响内部复位的操作。

看门狗操作通过置位 AUXR1 中的 WDE 位使能。一旦 WDE 置位后，只能通过复位来禁止。

如何操作看门狗定时器：

看门狗定时器的重装周期必须小于已编程的看门狗间隔。否则看门狗将溢出并产生系统复位。用户程序必须持续执行重装看门狗定时器的代码。两次执行之间的时间间隔必须小于看门狗间隔。当使用 12MHz 振荡器时，看门狗间隔可编程为 1024μs~261ms。

为了准备好看门狗操作的软件，程序员应当首先确定系统在错误的处理器状态中能维持多久。这个结果就是最大的看门狗间隔。当最大看门狗间隔变得更短时，要确保总是能在看门狗间隔之内重装看门狗定时器对程序员来说将变得更困难，看门狗操作的执行也变得更困难。

程序员必须以这样一种方式将软件分割开，即加入执行看门狗的重装以符合上述要求。程序员必须确定所有软件模块的执行时间。可能的条件分支、子程序、外部和内部中断都必须考虑到。由于有些代码的执行时间可能很难估算，这样就必须采用最坏的估计。任何情况下，程序员都必须确保在正常操作过程中看门狗不会被激活。

看门狗的重装分为两个阶段，这样可防止错误的软件对看门狗进行重装。首先必须置位 PCON.4(WLE)，然后可对 T3 进行装载，当 T3 装载后，PCON.4(WLE)自动复位。PCON.4(WLE)复位后 T3 将不能被装载。由于重装代码被频繁调用，可将其写成一个子程序。由于 T3 递增计数，因此装入 00H 将得到最大看门狗间隔，0FFH 将得到最小看门狗间隔。

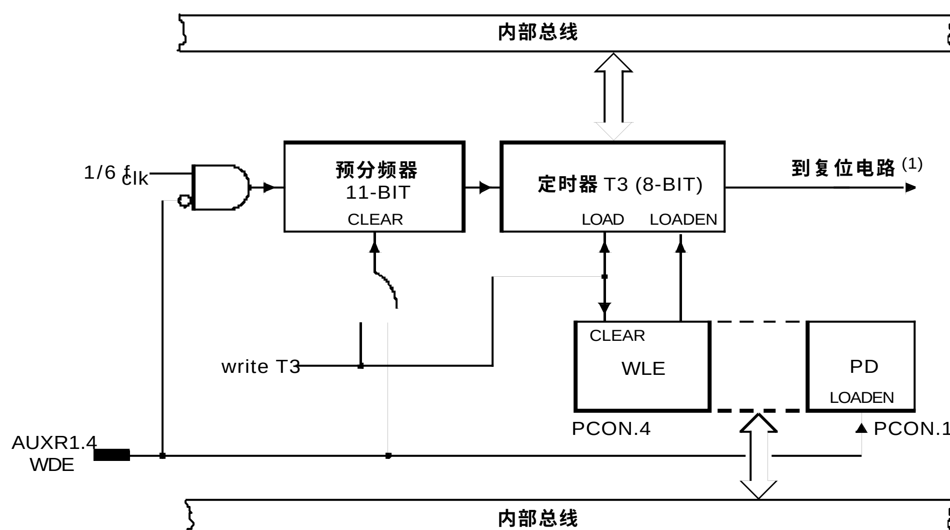
在空闲模式中，看门狗电路保持有效。当看门狗操作执行时不能使用掉电模式，因为这两种状态是冲突的。这样，当看门狗操作通过 AUXR1 中的 WDE 置位使能时，就无法进入掉电模式，对掉电模式位（PCON.1）的置位将会无效，PCON.1 仍然保持逻辑 0。

看门狗软件示例：

下面的示例指示了在用户程序中如何处理看门狗的操作。

```
;程序开始：
T3              EQU     0FFH            ;看门狗定时器 T3 的地址
PCON            EQU     087H            ;PCON 寄存器地址
WATCH-INTV      EQU     156             ;看门狗间隔（例如,2×100ms）
;插入到用户程序中每一个需要的位置
LCALL     WATCHDOG
;看门狗服务程序
WATCHDOG:       ORL     PCON,#10H       ;置位条件标志（PCON.4）
                MOV     T3,WATCH-INV;将看门狗间隔装入 T3
                RET
```

如果该子程序有可能在一个错误的状态中被调用，那么条件标志必须在主程序的不同部分置位。



(1) 见图8

图 46 T3 看门狗定时器功能框图

18 脉宽调制输出

PP8xC591 包含了两路脉宽调制输出（PWM）通道（见图 47）。这两路通道产生可编程宽度和间隔的脉冲。重复的频率由一个 8 位预分频器 PWMP 定义，PWMP 提供计数器的时钟。两路 PWM 共用预分频器和计数器。8 位计数器以 255 为模计数，即从 0 计数到 254。8 位计数器的值与两个寄存器 PWM0 和 PWM1 进行比较。

如果 PWM0 或 PWM1 的值大于计数器的值，对应的 PWM0 或 PWM1 输出置低电平。如果小于等于计数器的值，输出为高电平。因此占空比由 PWM0 和 PWM1 决定。占空比的范围为 0/255~255/255 并可以 1/255 的分辨率进行编程。

带缓冲的 PWM 输出可驱动直流电机。电机的转速与 PWMn 的值成比例。PWM 输出也可配置为一个双 DAC。

在此应用中，PWM 输出必须使用传统的运放电路进行积分。如果要得到精确的输出电压，应当在对 PWM 输出积分之前，使用带有独立模拟电源的外部缓冲器对 PWM 输出进行缓冲。

PWMn 输出脚 f_{PWM} 的重复频率由下式得到

$f_{PWM} = f_{CLK}/[(PWMP+1) \times 255]$

重复频率的范围为 184Hz~47kHz（f_{CLK}=12MHz 时）。将 00H 或 FFH 装入 PWM 寄存器，PWM 将分别输出恒定的高或低电平。由于 8 位计数器以 255 为模计数。，当 PWM 寄存器装入 FFH 时，计数器实际上永远达不到该值。

当比较寄存器（PWM0 或 PWM1）装入一个新值时，对应的输出立即更新。不需要等待当前的计数周期结束。两个 PWM 输出管脚都以推挽模式驱动。这两个管脚不作其它任何用途。

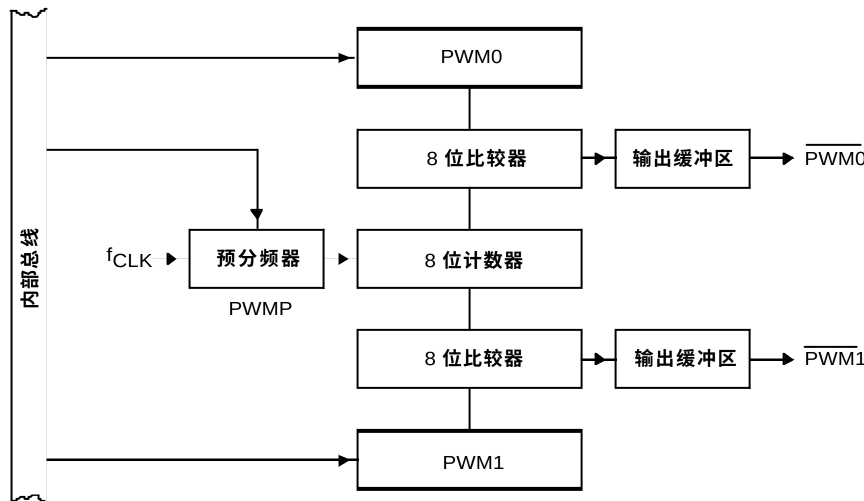


图 47 脉宽调制输出的功能框图

18.1 预分频器频率控制寄存器（PWMP）

读取 PWMP 得到当前重装值。预分频器的实际计数不能被读出。

表 82 预分频器频率控制寄存器（地址 FEH），复位值=00H

7	6	5	4	3	2	1	0
PWMP.7	PWMP.6	PWMP.5	PWMP.4	PWMP.3	PWMP.2	PWMP.1	PWMP.0

表 83 PWMP 位描述

位	符号	描述
7～0	PWMP.7 到 PWMP.0	分频系数：预分频器分频系数=（PWMP）+1

表 84 脉宽寄存器 0（地址 FCH），复位值=00H

7	6	5	4	3	2	1	0
PWM0.7	PWM0.6	PWM0.5	PWM0.4	PWM0.3	PWM0.2	PWM0.1	PWM0.0

表 85 PWM0 位描述

位	符号	描述
7～0	PWM0.7 到 PWM0.0	占空比：PWM0 信号低/高比率=(PWM0)/[255-(PWM0)]

表 86 脉宽寄存器 1（地址 FDH），复位值=00H

7	6	5	4	3	2	1	0
PWM1.7	PWM1.6	PWM1.5	PWM1.4	PWM1.3	PWM1.2	PWM1.1	PWM1.0

表 87 PWM1 位描述

位	符号	描述
7～0	PWM1.7 到 PWM1.0	占空比：PWM1 信号低/高比率=(PWM1)/[255-(PWM1)]

19 P1 口的操作

P1 口最多可输入 6 路 ADC 模拟信号。未用的 ADC 输入口可用于输入数字信号。这些输入有一个固有的滞后以防止

20 模－数转换器（ADC）

20.1 ADC 特性

- 10 位分辨率
- 6 路复用的模拟输入
- 由软件或外部信号启动转换
- 一次 10 位模数转换的转换时间：25µs @12MHz
- 差分非线性(DLe)：±1LSB
- 积分非线性(ILe)：±2LSB
- 偏移误差(OSe)：±2LSB
- 增益误差(Ge)：±4%
- 绝对电压误差(Ae)：3LSB
- 通道－通道匹配(Mctc)：±1LSB
- 模拟输入间串扰(Ct)：<60dB@100kHz
- 单调且无丢失代码
- 独立的模拟(V_{SSA})和数字(V_{DD},V_{SS})电源电压
- 参考电压特殊脚：$V_{ref(p)(A)}$

ADC 特性参考第 24 章

20.2 ADC 功能描述

模拟输入电路包括一个 6 输入模拟多路复用器和一个 10 位标准二进制逐次逼近式 ADC。A/D 还可通过置位 ADC8（AUXR1.7）配置成快速转换的 8 位模式。8 位结果保存在 ADCH 寄存器中。模拟参考电压和模拟电源通过单独的输入脚连接。对于 10 位精度，转换需要 50 个机器周期，即 25µs @12MHz。对于 8 位精度，转换需要 24 个机器周期。输入电压范围为 0~+5V。由于内部 DAC 带有一个比例分压计，在转换器特性中没有间断。图 48 所示为模拟输入电路的功能框图。

ADC 可选择在空闲模式中关闭以降低功耗或在空闲模式中保持有效以降低转换时的内部噪声。该选项可通过 AUXR1 中的 AIDL 位进行选择。当 AIDL 置位时，ADC 在空闲模式中有效，AIDL 清零时，ADC 在空闲模式中关闭。

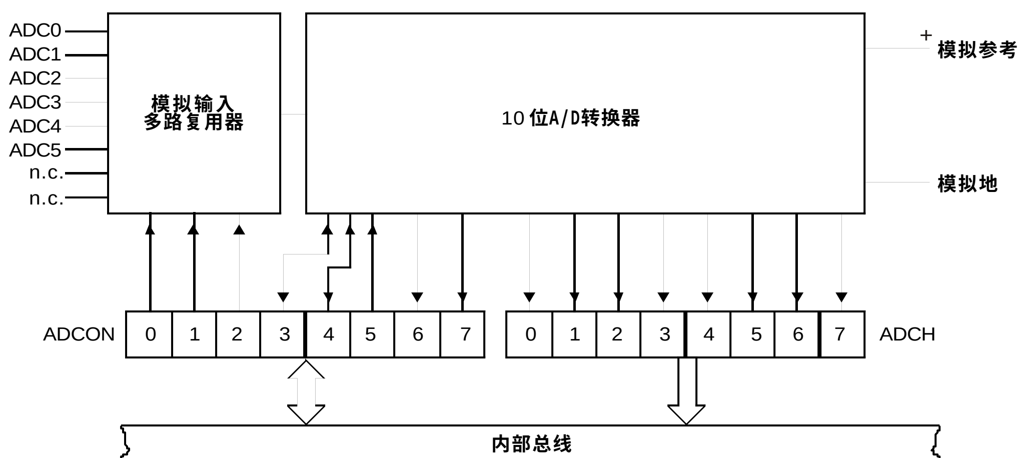


图 48 模拟输入电路的功能框图

20.3 10 位 A/D 转换

图 48 所示为逐次逼近式（SA）ADC 的部件。ADC 包含一个 DAC 将逐次逼近寄存器值转换成电压 V_{DAC} 与模拟输入电压 Vin 比较。比较器输出输入到逐次逼近控制逻辑对逐次逼近寄存器进行控制。通过置位 ADCON 中的 ADCS 位对转换进行初始化。ADCS 只能由软件置位。

当控制位 ADCON.5(ADEX)＝0 时，选择软件启动模式。通过置位控制位 ADCON.3(ADCS)启动转换。

当转换初始化完成时，转换在置位 ADCS **指令的下一个机器周期启动。**ADCS **实际上是作为两个触发器：一个受置位操作影响命令触发器和由读操作进行访问的状态标志。接下来的两个机器周期用于初始化转换器。在第一个周期的结束，**ADCS **状态标志置位，如果在转换过程中读取** ADCS **将返回** 1**。在第二个周期的结束开始对模拟输入进行采样。**

在接下来的 8 **个周期中，对之前选择的** P1 **口管脚的电压进行采样，输入电压应当保持稳定以获得有用的采样。在任何情况下，输入电压的变化率必须小于** 10V/ms **以防止出现一个不确定的值。**

逐次逼近控制逻辑首先置位逐次逼近寄存器的最高位并清除其它位（10 0000 0000B**）。**DAC **的输出（满度的** 50%**）与输入电压** V_{IN} **进行比较。如果输入电压大于** VDAC**，那么最高位保持置位，否则被清零。**

逐次逼近逻辑开始置位次高位（根据之前的结果取值 11 0000 0000B **或** 01 0000 0000B**），**V_{DAC}**再次与** V_{IN} **进行比较。如果输入电压大于** V_{DAC}**，那么被测试的位保持置位，否则清零。该过程一直重复到** 10 **个位都完成测试，此时转换的结果保存在逐次逼近寄存器中。图** 50 **所示为转换的流程图。位指针识别被测试的位。每个位花费** 4 **个机器周期。**

10 **位转换完成时由控制位** ADCON.4(ADCI)**标记。高** 8 **位结果保存在寄存器** ADCH **剩下** 2 **位保存在** ADCON.7(ADC.1)**和** ADCON.6(ADC.0)**中。用户可忽略** ADCON **中最低两位将** ADC **作为一个** 8 **位转换器（**ADCH **中的** 8 **位）。在任何情况下，整个实际转换的时间为** 50 **个机器周期。在命令触发器（**ADCS**）置位的** 50**（或** 24**）个周期之后** ADCI **将置位而** ADCS **状态标志将复位。**

控制位 ADCON.0,ADCON.1 **和** ADCON.2 **用于控制一个模拟多路复用器以选择** 6 **个模拟输入通道之一（见** 20.3.1**）。正在处理中的** ADC **转换不会被新的软件** ADC **启动所影响。已完成的转换结果在** ADC=1 **时保持不变。当进入空闲模式时，正在处理的** ADC **转换中止而已完成的转换结果保持不变。**

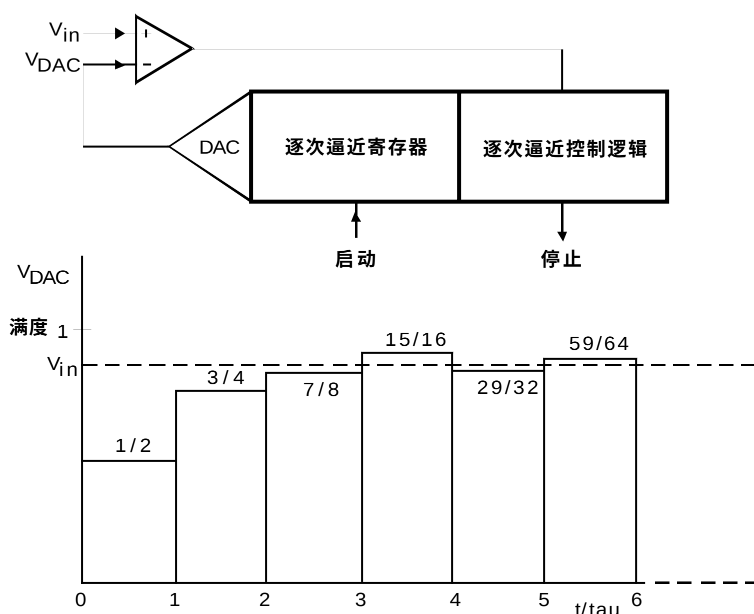


图 49 **逐次逼近式** ADC

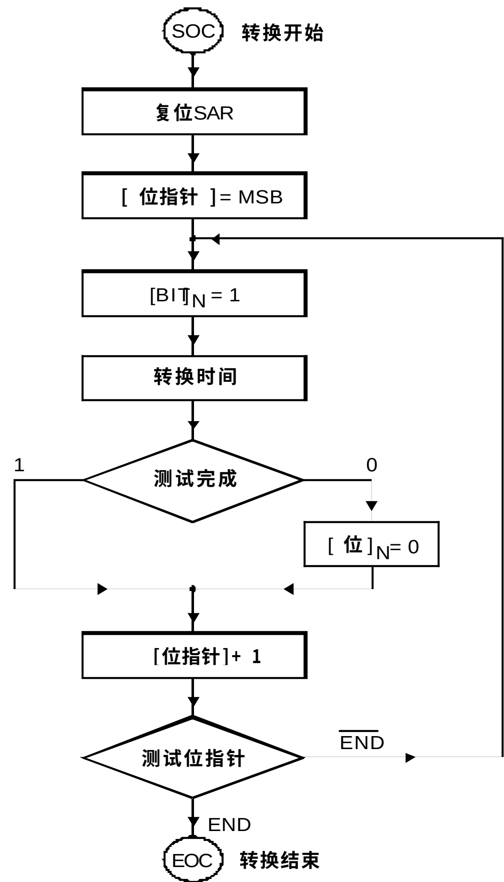


图 50 A/D 转换流程图

20.3.1 ADC 控制寄存器（ADCON）

表 88 ADC 控制寄存器（地址 C5H）；复位值＝xx00 0000B

7	6	5	4	3	2	1	0
ADC.1	ADC.0	ADEX	ADCI	ADCS	AADR2	AADR1	AADR0

表 89 ADCON 位描述

位	符号	描述
7	ADC.1	ADC 结果位 1
6	ADC.0	ADC 结果位 0
5	—	保留将来之用
4	ADCI	ADC 中断标志 该标志在 A/D 转换结果可读出时置位。如果中断使能则调用中断。该标志可由中断服务程序清零。当该标志置位时，ADC 不能启动新转换。ADCI 不能由软件置位
3	ADCS	ADC 启动和状态 置位该位启动一次 A/D 转换。它由软件置位。ADC 逻辑在 ADC 处理时确保该信号为高。转换结束时，ADCS 在中断标志置位后立即复位。ADCS 不能由软件复位。当 ADCS 或 ADCI 为高时（见表 90），不能启动新的转换 如果 ADCI 由软件清零同时 ADCS 置位，同一个通道的 A/D 转换可能会启动。但是建议在置位 ADCS 之前复位 ADCI。
2~0	AADR2~ AADR0	模拟输入选择 该二进制代码选择 P1 口 6 个模拟输入其中的一个作为 A/D 转换器的输入。只有当 ADCI 和 ADCS 都为 0 时才能改变。

表 90 ADC 状态

ADCI	ADCS	ADC 状态
0	0	ADC 空闲；可启动一次转换
0	1	ADC 忙；不能启动新的转换
1	0	转换完成；需要将 ADCI 清零以启动新转换
1	1	转换完成；需要将 ADCI 清零以启动新转换

表 91 选择模拟通道

AADR2	AADR1	AADR0	选择的模拟通道
0	0	0	ADC0(P1.2)
0	0	1	ADC1(P1.3)
0	1	0	ADC2(P1.4)
0	1	1	ADC3(P1.5)
1	0	0	ADC4(P1.6)
1	0	1	ADC5(P1.7)

20.4 10 位 ADC 分辨率和模拟电源

图 51 所示为 ADC 的实现。ADC 具有自己的模拟地（AVss），正向模拟参考管脚 Vref+连接到每个 DAC 的电阻梯的末端。电阻梯有 1023 个相等间距的抽头，间隔的电阻值为“R”。第一个抽头位于 AVss 之上的 0.5R 处，最后一个抽头位于 Vref+以下 1.5R 处。这就使整个电阻梯的阻值为 1024R。该结构确保了 DAC 为单调的并且量化误差为对称的，如图 53 所示。

对于 0V 到+1/2LSB 之间的输入电压，10 位 A/D 转换的结果为 00 0000 0000B＝0000H。对于 (Vref+)-3/2LSB 到 Vref+之间的输入电压，转换的结果为 11 1111 1111B＝3FFFH。AVref+的值可在 V_{DD}+0.2V 和 AVss-0.2V 之间。AVref+应当为正的 0V 和 AVref+。如果模拟输入电压的范围为 2V~4V，在 AVref+＝4V 时可获得 10 位分辨率。结果可通过下面的公式得到：

结果＝$1024 \times V_{IN}$/ AVref+

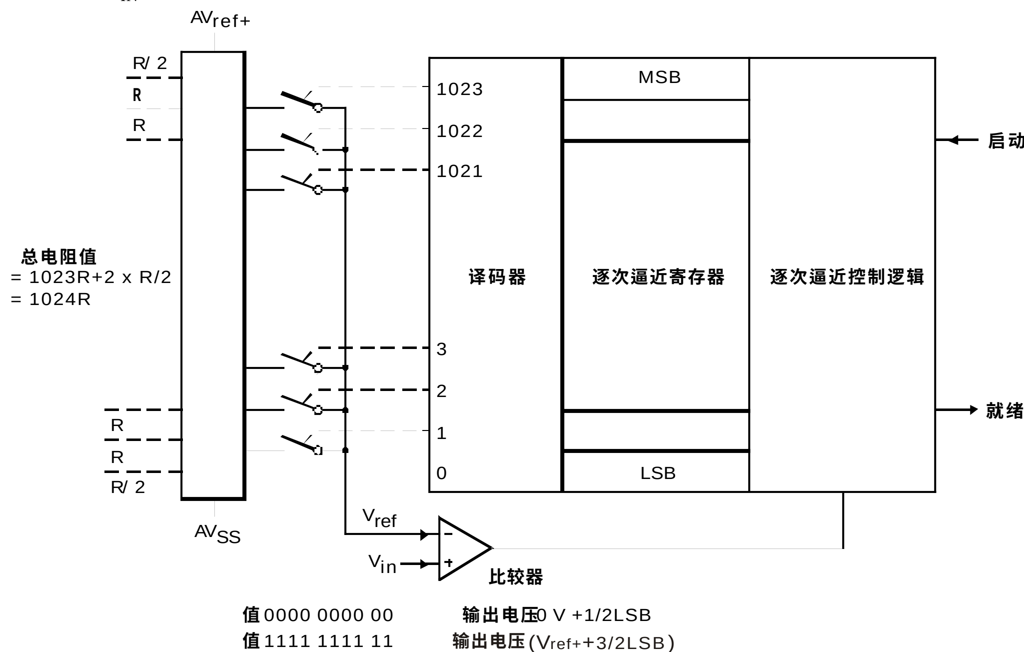
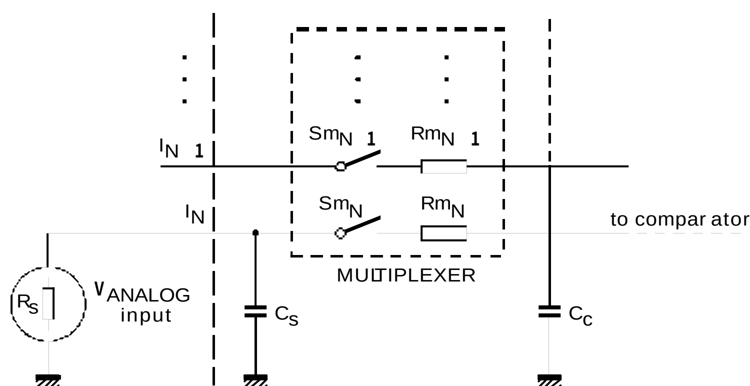


图 51 ADC 实现



Rm = 0.5 - 3 k

$C_S + C_C$ = 15 pF maximum

R_S = Recommended < 9.6 k for 1 LSB @ 12 MHz

图 52 A/D 输入：等效电路

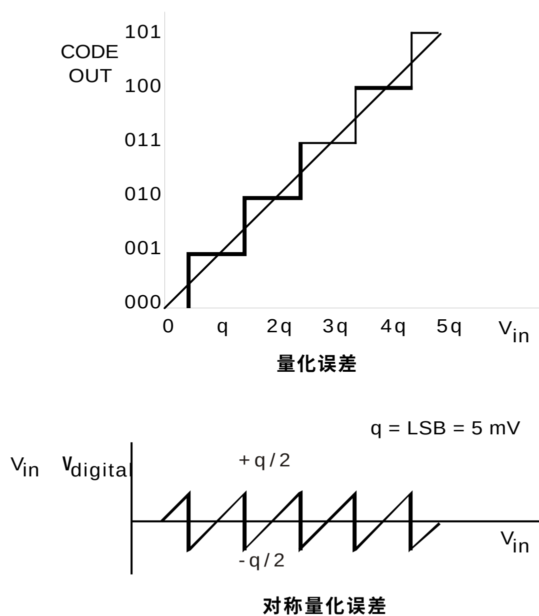


图 53 有效转换特性

20.5 节电模式

PP8xC591 有两种节电模式：空闲模式和掉电模式。通过置位 PCON 寄存器中的相应位进入这两种模式。当 PP8xC591 进入空闲模式时，下面的功能被禁止：

CPU	暂停
定时器 T2	暂停并复位
PWM0,PWM1	复位：输出为高
ADC	可通过置位 AIDC(AUXR1.6)在空闲模式下工作

在空闲模式中，下面的功能保持有效：

定时器 0

定时器 1

定时器 T3

SIO0,SIO1

外部中断

当 PP8xC591 进入掉电模式时，振荡器停振。通过置位 PCON 寄存器中的 PD 位进入。PD 位只有在 WD 位为 0 时才能置位。

21 中断

PP8xC591 有 15 个中断源，每一个都可分配为 4 个中断优先级之一。5 个与 80C51 相同的中断源分别为外部中断（INT0 和 INT1），定时器 0 和定时器 1 中断（IT0 和 IT1）和串口中断（RI 或 TI）。在 PP8xC591 中，标准的串口中断称为 SIO0。

7 个定时器 T2 中断分别由标志 CTI0~CTI3，CMI0~CMI1 和标志 T2OV 和 T2BO 的逻辑或产生。标志 CTI0~CTI3 由输入信号 CT0I~CT3I 置位。如果不使用 T2 的捕获功能，输入 INT2 到 INT5 可看作是 4 个额外的外部中断（详见 16.1.4.1）。

当定时器 T2 和比较寄存器 CM0 和 CM1 之间发生匹配时，标志 CMI0~CMI1 置位。当发生 8 位或 16 位溢出时标志 T2BO 和 T2OV 分别置位。这 8 个标志不由硬件而是由软件清零以避免连续产生中断。

ADC 中断由 ADC 控制寄存器（ADCON）中的 ADCI 标志产生。当 ADC 转换结果可读出时该标志置位。ADCI 不由硬件而是由软件清零以避免连续产生中断。SIO1（I^2C）中断由 SIO1 控制寄存器（S1CON）中的 SI 标志产生。当 S1STA 装入有效的状态代码时，该标志置位。

ADCI 标志可通过软件复位但不能由软件置位。所有其它产生中断的标志都可通过软件置位或清零，其效果与通过硬件置位或清零相同。因此，可通过软件产生中断，而挂起的中断也可通过软件取消。

当 CANCON 寄存器中的一个或多个位置位时产生 CAN 中断（向量地址 006BH）（详见 12.5.5）。

21.1 中断使能寄存器

每个中断源都可通过置位或清零中断使能寄存器 IEN0 和 IEN1 中的相应位单独使能或禁止。通过置位或清零 IEN0 中的 EA 位，可全局使能或禁止所有的中断源（中断使能寄存器的描述见 21.2.1 和 21.2.2）。

3 个寄存器与 4 优先级中断之一相关。它们分别为 IENx，IPx 和 IPxH（见 21.2.3 和 21.2.6）。IPxH（中断优先级高）寄存器使 4 优先级中断结构成为可能。

IPxH 的功能很简单，当它与 IPx 组合使用时决定每个中断源的优先级。如下表所示：

表 92 中断优先级寄存器

优先级位		中断优先级
IPxH.x	IPx.x	
0	0	0（最低优先级）
0	1	1
1	0	2
1	1	3（最高优先级）

中断服务优先级的配置与 80C51 相同，区别在于 PP8xC591 有 4 个中断优先级而 80C51 为 2 个。如果当前没有同级或更高级的中断处于服务中，中断可以被执行。如果同级或更高级的中断在处理中，新的中断必须等待直到当前中断执行完毕。如果较低级的中断正在处理，它将会停止转而执行新的中断。当新的中断执行完毕，较低级的中断才可继续。

21.2 中断使能和优先级寄存器

21.2.1 中断使能寄存器 0（IEN0）

逻辑 0＝中断禁止；逻辑 1＝中断使能

表 93 中断使能寄存器 0（地址 A8H）

7	6	5	4	3	2	1	0
EA	EAD	ES1	ES0	ET1	EX1	ET0	EX0

表 94 IEN0 位描述

位	符号	描述
7	EA	全局使能/禁止位 如果 EA=0，所有的中断都被禁止;EA=1 时，所有的中断都可通过设置/清零各自的使能位单独使能或禁止。
6	EAD	ADC 中断使能位
5	ES1	SIO1（I^2C）中断使能位
4	ES0	SIO0（UART）中断使能位
3	ET1	定时器 1 中断使能位
2	EX1	外部中断 1/第二中断使能位
1	ET0	定时器 0 中断使能位
0	EX0	外部中断 0 使能位

21.2.2 中断使能寄存器 1（IEN1）

逻辑 0＝中断禁止；逻辑 1＝中断使能

表 95 中断使能寄存器 1（地址 E8H）

7	6	5	4	3	2	1	0
ET2	ECAN	ECM1	ECM0	ECT3	ECT2	ECT1	ECT0

表 96 IEN1 位描述

位	符号	描述
7	ET2	T2 溢出中断使能位
6	ECAN	CAN 中断使能位
5	ECM1	T2 比较器 1 中断使能位
4	ECM0	T2 比较器 0 中断使能位
3	ECT3	T2 捕获寄存器 3 中断使能位
2	ECT2	T2 捕获寄存器 2 中断使能位
1	ECT1	T2 捕获寄存器 1 中断使能位
0	ECT0	T2 捕获寄存器 0 中断使能位

21.2.3 中断优先级寄存器 0（IP0）

逻辑 0＝低优先级；逻辑 1＝高优先级

表 97 中断优先级寄存器 0（地址 B8H）

7	6	5	4	3	2	1	0
—	PAD	PS1	PS0	PT1	PX1	PT0	PX0

表 98 IP0 位描述

位	符号	描述
7	—	保留位
6	PAD	ADC 中断优先级位
5	PS1	SIO1（I^2C）中断优先级位
4	PS0	SIO0（UART）中断优先级位
3	PT1	定时器 1 中断优先级位
2	PX1	外部中断 1/第二中断优先级位
1	PT0	定时器 0 中断优先级位
0	PX0	外部中断 0 优先级位

21.2.4 中断优先级高寄存器 0（IP0H）

逻辑 0＝低优先级；逻辑 1＝高优先级

表 99 中断优先级高寄存器 0（地址 B7H）

7	6	5	4	3	2	1	0
—	PADH	PS1H	PS0H	PT1H	PX1H	PT0H	PX0H

表 100 IP0H 位描述

位	符号	描述
7	—	保留位
6	PADH	ADC 中断优先级位
5	PS1H	SIO1（I^2C）中断优先级位
4	PS0H	SIO0（UART）中断优先级位
3	PT1H	定时器 1 中断优先级位
2	PX1H	外部中断 1/第二中断优先级位
1	PT0H	定时器 0 中断优先级位
0	PX0H	外部中断 0 优先级位

21.2.5 中断优先级寄存器 1（IP1）

逻辑 0＝低优先级；逻辑 1＝高优先级

表 101 中断优先级寄存器 1（地址 F8H）

7	6	5	4	3	2	1	0
PT2	PCAN	PCM1	PCM0	PCT3	PCT2	PCT1	PCT0

表 102 IP1 位描述

位	符号	描述
7	PT2	T2 溢出中断优先级位
6	PCAN	CAN 中断优先级位
5	PCM1	T2 比较器 1 中断优先级位
4	PCM0	T2 比较器 0 中断优先级位
3	PCT3	T2 捕获寄存器 3 中断优先级位
2	PCT2	T2 捕获寄存器 2 中断优先级位
1	PCT1	T2 捕获寄存器 1 中断优先级位
0	PCT0	T2 捕获寄存器 0 中断优先级位

21.2.6 中断优先级高寄存器 1（IP1H）

逻辑 0＝低优先级；逻辑 1＝高优先级

表 103 中断优先级高寄存器 1（地址 F7H）

7	6	5	4	3	2	1	0
PT2H	PCANH	PCM1H	PCM0H	PCT3H	PCT2H	PCT1H	PCT0H

表 104 IP1H 位描述

位	符号	描述
7	PT2H	T2 溢出中断优先级位
6	PCANH	CAN 中断优先级位
5	PCM1H	T2 比较器 1 中断优先级位
4	PCM0H	T2 比较器 0 中断优先级位
3	PCT3H	T2 捕获寄存器 3 中断优先级位
2	PCT2H	T2 捕获寄存器 2 中断优先级位
1	PCT1H	T2 捕获寄存器 1 中断优先级位
0	PCT0H	T2 捕获寄存器 0 中断优先级位

21.3 中断优先级

表 105 中断优先级结构

中断源	符号	优先级
外部中断 0	X0	最高
SIO1(I2C)	S1	↑
ADC 完成	ADC	
定时器 0 溢出	T0	
T2 捕获 0	CT0	
T2 比较 0	CM0	
外部中断 1	X1	
T2 捕获 1	CT1	
T2 比较 1	CM1	
定时器 1 溢出	T1	
T2 捕获 2	CT2	
CAN	CAN	
串口 0(UART)	S0	
T2 捕获 3	CT3	↓
定时器 T2 溢出	T2	最低

21.4 中断向量

向量指示相应中断服务程序入口在程序存储器中的位置。

表 106 中断向量地址

中断源	符号	向量
外部中断 0	X0	0003H
定时器 0 溢出	T0	000BH
外部中断 1	X1	0013H
定时器 1 溢出	T1	001BH
串口 0(UART)	S0	0023H
SIO1(I2C)	S1	002BH
ADC 完成	ADC	0033H
T2 捕获 0	CT0	003BH
T2 捕获 1	CT1	0043H
T2 捕获 2	CT2	004BH
T2 捕获 3	CT3	0053H
T2 比较 0	CM0	005BH
T2 比较 1	CM1	0063H
CAN	CAN	006BH
定时器 T2 溢出	T2	0073H

23 极限参数

符号	参数	最小值	最大值	单位
V_{DD}	VDD 对地和 SCL，SDA 对 Vss 电压	-0.5	+6.5	V
V_I	其它任何脚相对于 Vss 的电压	-0.5	V_{DD}+0.5	V
I_I,I_O	每个 I/O 脚的最大输入/输出电流	—	5	mA
Ptot	总的功率损耗	—	1.0	W
Tstg	贮存温度范围	-65	+150	℃
Tamb	操作温度	-40	+85	℃
Vpp	EA/Vpp 脚相对于 Vss 的电压	-0.5	+13	V

注：

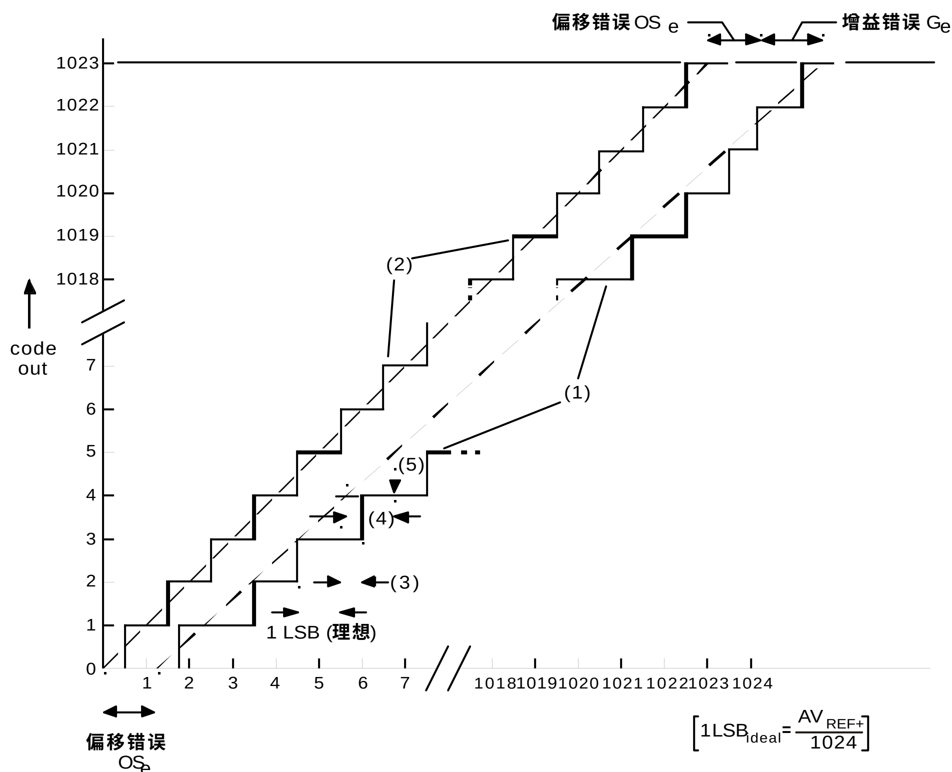
1. 器件在超过上面所列的极限值情况下工作，可能会造成永久性的损坏。
2. 本产品有保护器件内部的电路设计，避免超负荷的损坏性影响。不过建议避免在超过最大值的情况下工作。
3. 参数在操作温度范围内是有效的，除非另有规定。所有的电压都是相对 Vss 而言的，除非另有说明。

24 DC 特性

符号	参数	测试条件	范围		单位
			最小	最大	
电源					
I_{DD}	电源电流,工作模式	t_{CLK}=12MHz	-	45	mA
I_{ID}	电源电流,空闲模式	t_{CLK}=12MHz	-	25	mA
I_{PD}	电源电流,掉电模式	2V<V_{PD}<V_{DD}	-	100	μA
输入					
V_{IL}	输入电压低电平 (P1.0,P1.1,P1.6,P1.7 除外)		-0.5	2V_{DD}-0.1	V
V_{IL1}	EA 输入电压低电平		-0.5	0.2V_{DD}-0.3	V
V_{IL2}	P1.0 和 P1.1 输入电压低电平		-0.5	0.2V_{DD}	V
V_{IL3}	P1.6 和 P1.7 输入电压低电平			0.3 V_{DD}	V
V_{IH}	输入电压高电平(P1.0,P1.1,P1.6, P1.7,XTAL1,RST 除外)		0.2V_{DD}+0.9	V_{DD}+0.5	V
V_{IH1}	XTAL1,RST 输入电压高电平		0.7 V_{DD}	V_{DD}+0.5	V
V_{IH2}	P1.6 和 P1.7 输入电压高电平		0.7 V_{DD}	6	V
V_{IH3}	P1.0 和 P1.1 输入电压高电平		0.8 V_{DD}	V_{DD}	V
I_{IL}	低电平输入电流（准双向口输出模式的 P1,P2 和 P3，P1.6,P1.7 除外）	V_{IN}=0.45V	-1	-50	μA
I_{TL}	逻辑 1 到 0 跳变电流（准双向口输出模式的 P1,P2 和 P3，P1.6,P1.7 除外）			-650	μA
I_{LI}	输入漏电流（高阻模式的 P0,P2,P3,P1.0 和 P1.1）	0.45V<V_{IN}<V_{DD}		±10	μA
I_{IL2}	P1 输入漏电流（P1.0 和 P1.1 除外）	0.45V<V_{IN}<V_{DD}		1	μA
V_{OL}	输出低电压 [5,9]	I_{OL}=3.2mA,V_{DD}=2.7V		0.4	V

输出					
V_{OL}	P1,P2,P3 低电平输出电压(P1.0, P1.6,P1.7 除外)	I_{OL}=1.6mA		0.4	V
V_{OL1}	P0,ALE,PSEN,RST,PWM0,PWM1 低电平输出电压	I_{OL}=3.2mA		0.4	V
V_{OL2}	P1.6 和 P1.7 低电平输出电压	I_{OL}=3.0mA		0.4	V
V_{OL3}	P1.0 和 P1.1 低电平输出电压	I_{OL}=8.0mA		0.3 V_{DD}	V
V_{OH}	P1,P2,P3 准双向输出模式时的高电平输出电压(P1.1,P1.6,P1.7 除外)	I_{OH}=-60μA	V_{DD}-0.7		V
V_{OH1}	高电平输出电压(P0 和 P2 外部总线模式,P2 推挽模式,ALE,PSEN, PWM0,PWM1)	I_{OH}=-3.2mA	V_{DD}-0.7		V
V_{OH2}	高电平输出电压(P1.0 和 P1.1)	I_{OH}=-1.6mA	0.7 V_{DD}		V
V_{OH3}	高电平输出电压(P1,P2,P3 推挽输出模式) (P1.0,P1.1,P1.6,P1.7 除外)	I_{OH}=-1.6mA	V_{DD}-0.7		V
R_{RST}	复位上拉电阻		40	225	kΩ
C_{IO}	输入/出口感应电容	测试频率=1MHz, Tamb=25℃	-	15	Pf
模拟输入					
AV_{IN}	模拟输入电压		AV_{SS}-0.2	V_{DD}+0.2	V
AV_{REF}	参考电压		-	V_{DD}+0.2	V
R_{REF}	AVref+和 AVss 之间的电阻		10	50	kΩ
C_{IA}	模拟输入电容		-	15	pF
t_{ADS}	采样时间		-	5tcy;(1) 8tcy	μs
t_{ADC}	转换时间		-	24tcy;(1) 50tcy	μs
DLe	差分非线性		-	±1	LSB
ILe8	积分非线性（8 位模式）		-	±1	LSB
ILe	积分非线性			±2	LSB
OSe8	偏移误差（8 位模式）			±1	LSB
OSe	偏移误差			±2	LSB
Ge	增益误差			±0.4	%
Ae	绝对电压误差			±3	LSB
Mctc	通道间匹配			±1	LSB
Ct	P1 口模拟输入间串扰	0～100kHz	-	-60	dB

注: (1) 8位模式



(1) 实际转换示例
(2) 理想的转换曲线
(3) 差分非线性(DLe)
(4) 积分非线性(ILe)
(5) 实际转换曲线中心轴线

图 54 ADC 转换特性

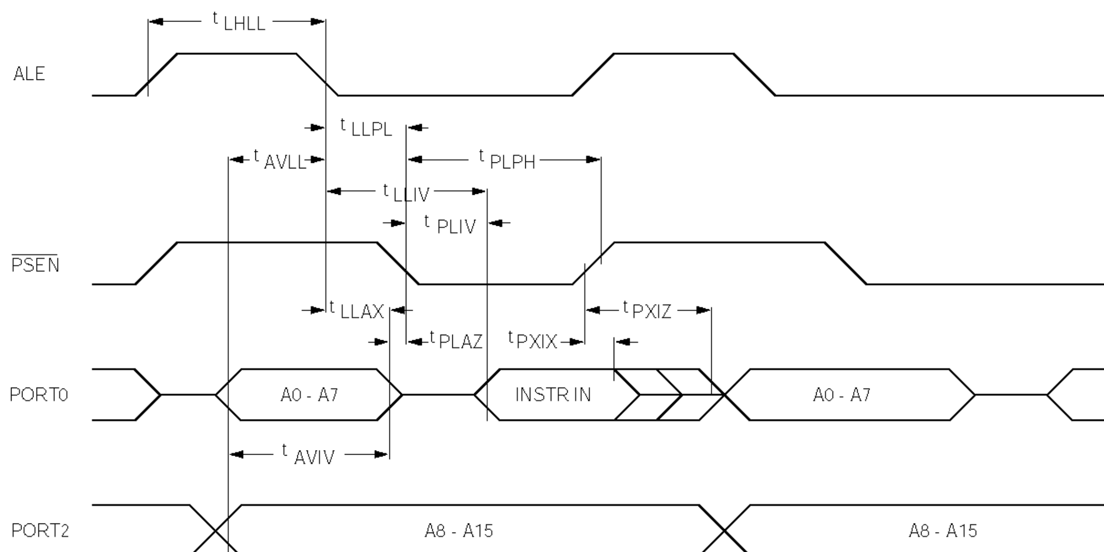


图 55 外部程序存储器读周期

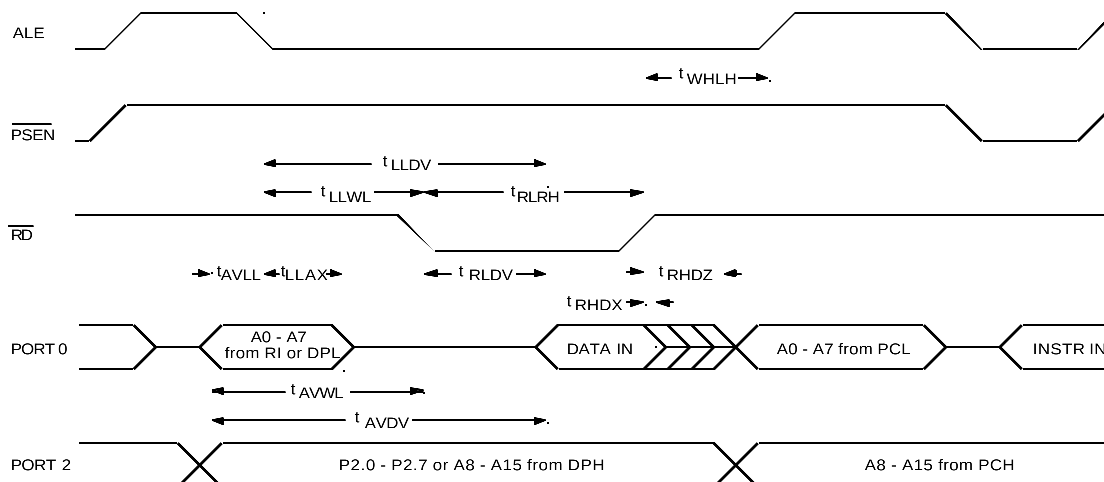


图 56 外部数据存储器读周期

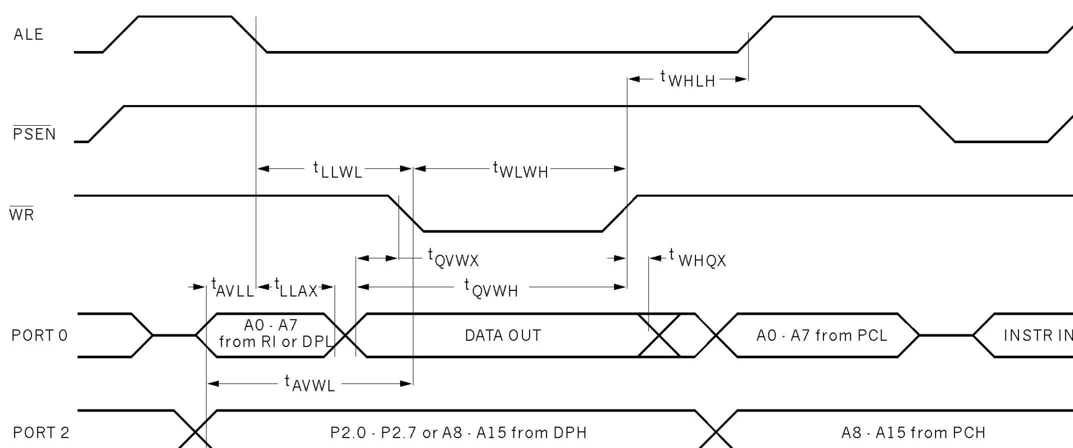


图 57 外部数据存储器写周期

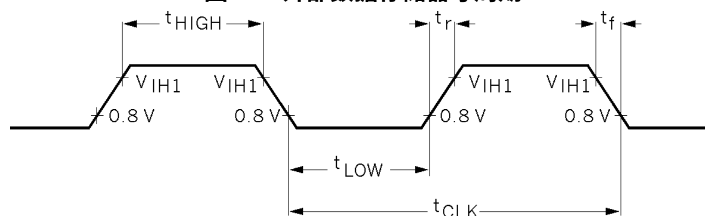


图 58 外部时钟驱动 XTAL1

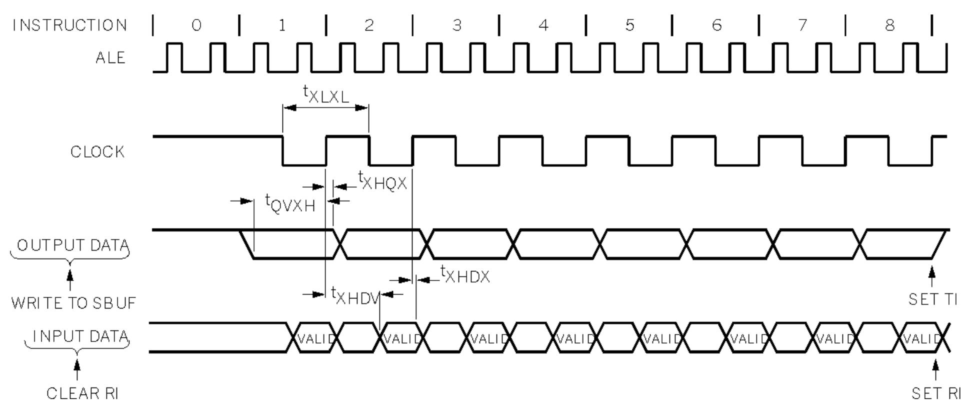


图 59 移位寄存器模式时序图

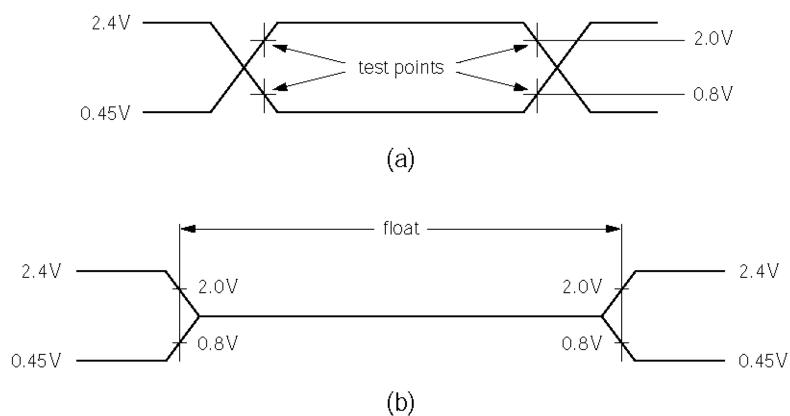


图 60　AC 测试输入,输出波形(a)和漂移波形(b)

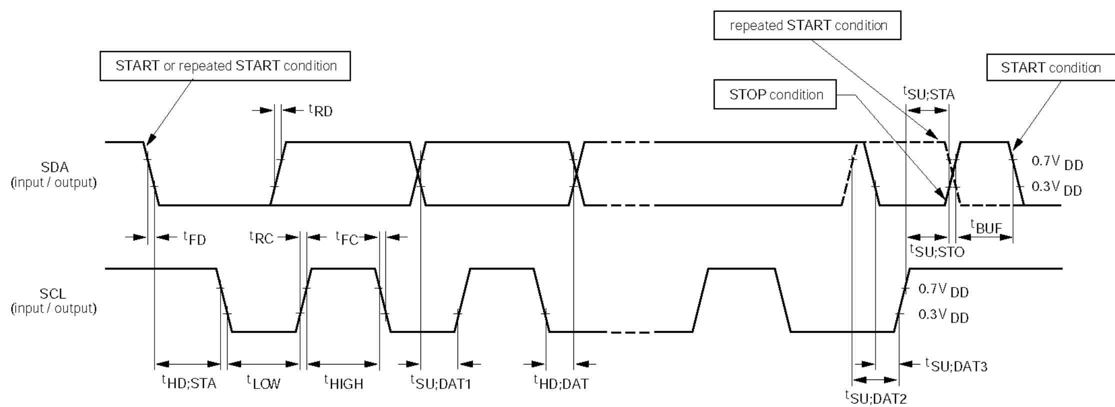


图 61　I²C 接口时序

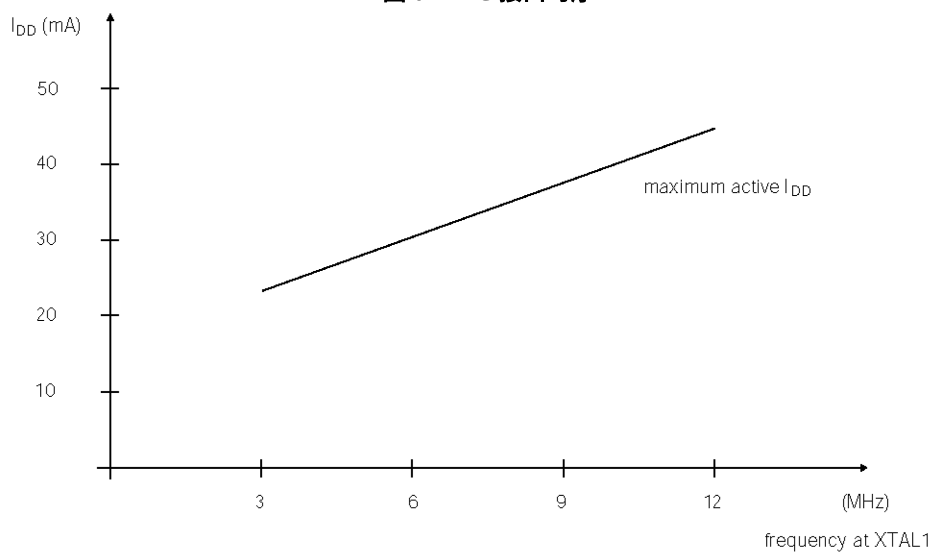
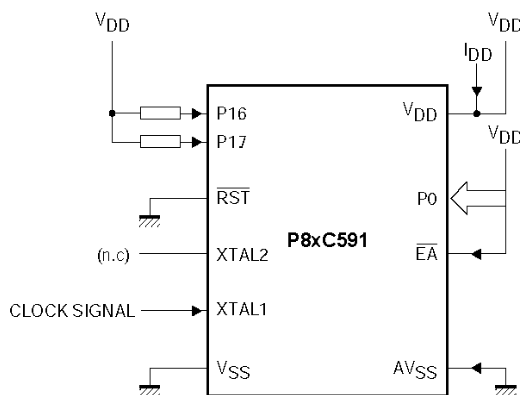
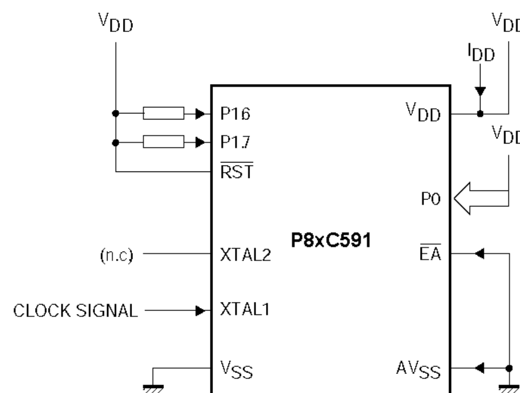


图 62　I_{DD} 与频率的函数关系



All other pins are disconnected.
(1) The following pins must be forced to V_{DD}: $\overline{EA}$ and Port 0.
(2) The following pins must be forced to V_{SS}: AV_{SS} and $\overline{RST}$.
(3) Port 1.6 and 1.7 should be connected to V_{DD} through resistors of sufficiently high value such that the sink current into these pins cannot exceed the I_{OL1} spec of the pins.
(4) The following pins must be disconnected: XTAL2 and all pins not specified above.
(5) Note, during reset = active the power consumption will be reduced by an internal clock divider by two.

图 63 I_{DD} 测试条件，工作模式



All other pins are disconnected.
(1) The following pins must be forced to V_{DD}: Port 0 and $\overline{RST}$.
(2) The following pins must be forced to V_{SS}: AV_{SS} and $\overline{EA}$.
(3) Port 1.6 and 1.7 should be connected to V_{DD} through resistors of sufficiently high value such that the sink current into these pins cannot exceed the I_{OL1} spec of the pins. These pins must not have logic 0 written to them prior to this measurement.
(4) The following pins must be disconnected: XTAL2 and all pins not specified above.

图 64 IDD 测试条件，空闲模式

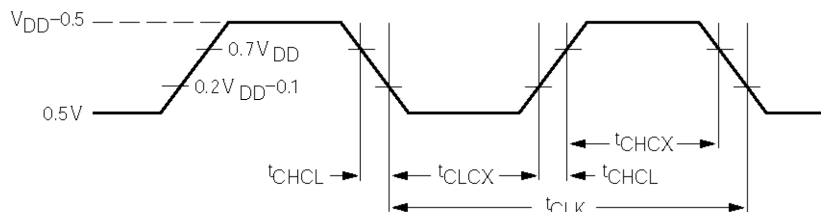
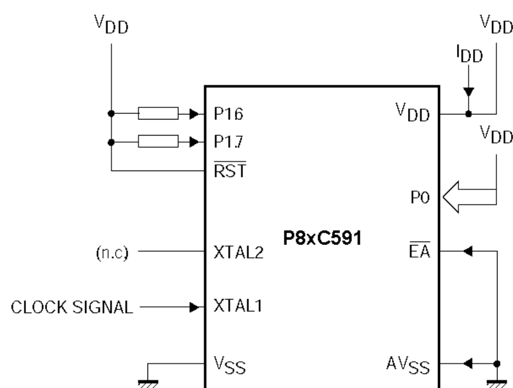


图 65 工作和空闲模式下用于 I_{DD} 测试的时钟波形 t_{CLCH}=t_{CHCL}=10ns



All other pins are disconnected. V_{DD} = 2 V to 5.5V

(1) The following pins must be forced to V_{DD}: Port 0 and $\overline{RST}$.

(2) The following pins must be forced to V_{SS}: AV_{SS} and $\overline{EA}$.

(3) Port 1.6 and 1.7 should be connected to V_{DD} through resistors of sufficiently high value such that the sink current into these pins cannot exceed the I_{OL1} spec of the pins. These pins must not have logic 0 written to them prior to this measurement.

(4) The following pins must be disconnected: XTAL2 and all pins not specified above.

图 66 IDD 测试条件，掉电模式

25.1 时序符号定义

振荡器：

fclk=时钟频率

tclk=时钟周期

时序符号（首字母缩写）

每个时序标号有 5 个特征。首先是“t”（时序），其他特征取决于他们的位置，用来表示信号名或信号的逻辑状态，说明如下：

A — 地址		P — PSEN	
C — 时钟		Q — 数据输出	
D — 输入数据		R — RD 信号	
H — 逻辑高电平		t — 时间	
I — 指令（编程存贮器的内容）		W — WR 信号	
L — 逻辑低电平或 ALE		X — 不再是有效逻辑电平	
Z — 悬浮		V — 有效	

例如：t_{AVLL}=从地址有效到 ALE 为低的时间。

t_{LLPL}=从 ALE 为低到 $\overline{PSEN}$ 为低的时间。

26 EPROM 特性

P8xC591 包括 3 个标识字节可通过 EPROM 编程系统读出，用于对器件进行识别。通过标识字节可以识别出器件是由 Philips 制造的。

(030H)=15H 表示由 Phlips 制造

(0031H)=98H 表示 Hamburg

(60H)=01H 表示 87C591

26.1 编程校验

如果保密位 2 和 3 还未被编程，片内程序存贮器的内容可以被读出以校验程序。P1 口和 P2 口提供所要读出的存贮器的地址（参见图 41），其它管脚保持在‘校验程序数据’电平上（参见表 8），地址的内容由 P0 口输出。在这个操作中，P0 口需外接上拉电阻。

如果密码表已被编程，P0 口的数据是程序字节与其中一个密码字节异或的结果。为了正确解码校验数

据，用户必须知道密码表的内容，而密码表的内容不能被读出。

26.2 保密位

如果任一个保密位都没被编程，则可以对程序存贮器的代码进行校验。密码表被编程后，当校验程序时，程序将被加密。当仅编程保密位 1 时（见表 113），MOVC 指令（访问外部程序存贮器时）被禁止从内部存贮器取代码字节，EA 脚由复位关闭，且所有 EPROM 编程被禁止。当保密位 1 和 2 同时被编程时，除了上面所述外，校验方式也被禁止。当编程了所有三个保密位时，除上面所述外，还禁止外部程序存贮器。

表 113 为 EPROM 器件编程保密位

编程锁位 [1, 2]				保护描述
	SB1	SB2	SB3	
1	U	U	U	没有保密特性使能
2	P	U	U	MOVC 指令(访问外部程序存贮器时)被禁止从内部存贮器取代码字节,EA 被采样并由复位关闭,EPROM 程序被禁止
3	P	P	U	同 2,加上校验方式被禁止
4	P	P	P	同 3,再加上外部程序存贮器被禁止

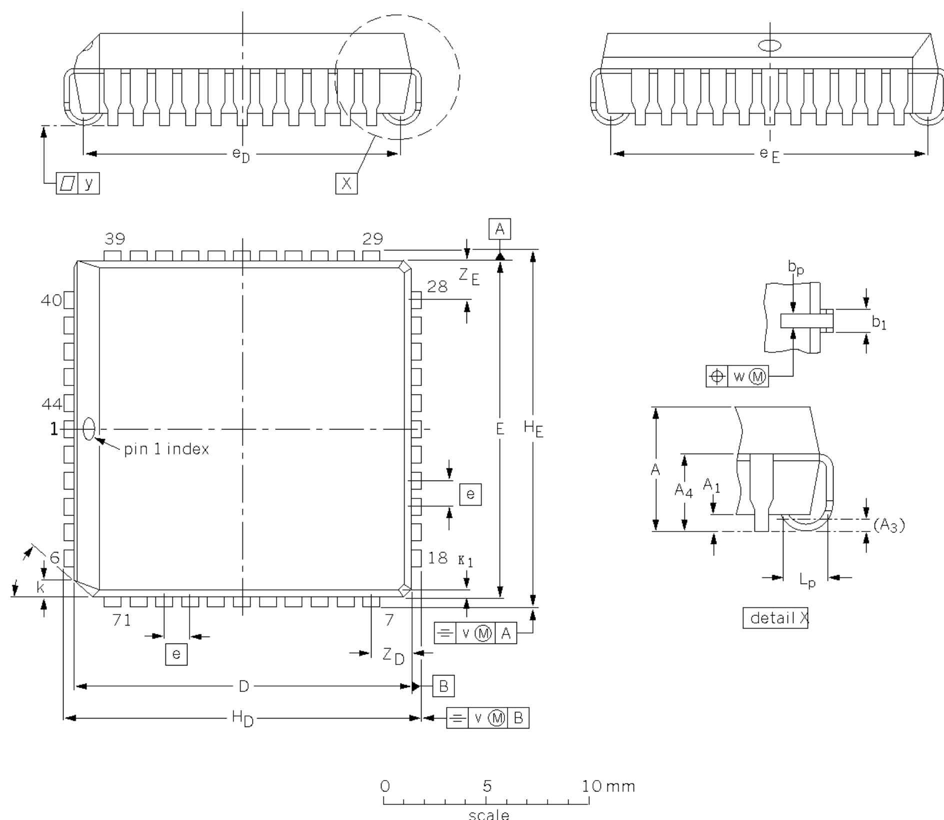
注:

1 P—已编程， U—未编程

2 其它的保密位的组合未定义

27 封装信息

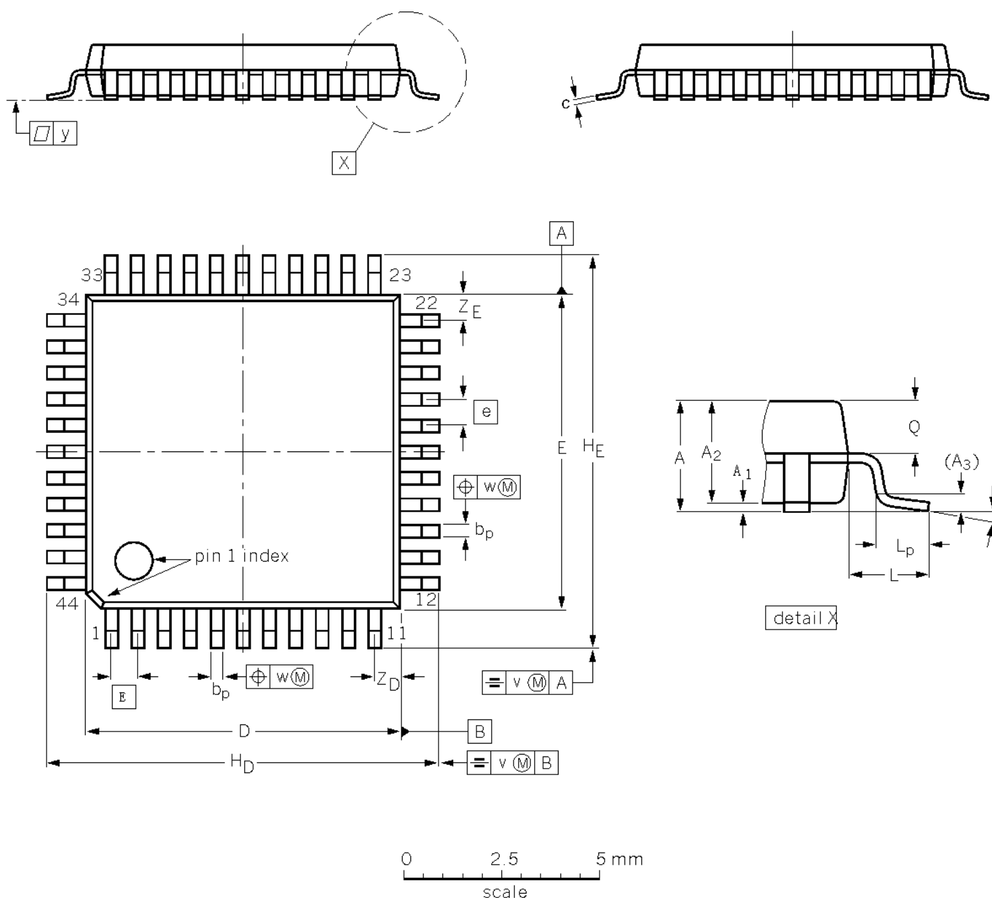
PLCC 44 脚



DIMENSIONS (MILLIMETRE DIMENSIONS ARE DERIVED FROM THE ORIGINAL INCH DIMENSIONS)

UNIT	A	A_1 min.	A_3	A_4 max.	b_p	b_1	D(1)	E(1)	eH	e_D	E_E	H_D	E	K	K_1 max.	L_p	v	w	y	Z_D(1) max.	Z_E(1) max.	
MM	4.57 4.19	0.51	0.25	3.05	0.53 0.33	0.81 0.66	16.66 16.51	16.66 16.51	1.27	16.00 14.99	16.00 14.99	17.65 17.40	17.65 17.40	1.22 1.07	0.51	1.44 1.02	0.18	0.18	0.10	2.16	2.16	45°
inches	0.180 0.165	0.020	0.01	0.12	0.021 0.013	0.032 0.026	0.656 0.650	0.656 0.650	0.05	0.630 0.590	0.630 0.590	0.695 0.685	0.695 0.685	0.048 0.042	0.020	0.057 0.040	0.007	0.007	0.004	0.085	0.085	

QFP44：44 脚；本体 10×10×1.75mm



DIMENSIONS (MM ARE THE ORIGINAL DIMENSIONS)

UNIT	A max.	A_1	A_2	A_3	b_p	cE	D(1)	(1)	eH	H_D	E	LL	p	QZ	v	w	y	Z_D(1)	E(1)	
MM	2.10	0.25 0.05	1.85 1.65	0.25	0.40 0.20	0.25 0.14	10.1 9.9	10.1 9.9	0.8	12.9 12.3	12.9 12.3	1.3	0.95 0.55	0.85 0.75	0.15	0.15	0.1	1.2 0.8	1.2 0.8	10° 0°

勘误：

偏差 1：CAN 复位模式

在该器件的 CAN 部分发现一个异常情况。该异常是指 CAN 复位模式（软件复位），外部（硬件）复位不受影响。

描述

用户软件在正常操作时进入复位模式或 CAN 控制器进入总线关闭状态时，会发生此异常情况。在该情况下，接收缓冲区和它和接收信息计数器和定义的一样复位，但是 RX FIFO 和发送缓冲区的内部指针没有正确复位。结果该 CAN 控制器的正常操作导致无法再重新进入操作模式。

解决办法

1） **强烈建议在进入 CAN 复位模式之前先清空 RX FIFO。可通过“释放接收缓冲区命令”执行。**

2） **不允许在发送或接收过程中进入 CAN 复位模式。在 PP8xC591 可以进入 CAN 复位模式之前必须终止挂起的发送（等待发送成功或放弃发送）。**

此两种情况的代码可类似于此：

```
do {
        // read message bytes
        Release Receive Buffer = 1; // empty RXFIFO
```

```
        } while (RBS == 1);
do {} while (TS); // wait until Transmit Status = 0
do {} while (RS); // wait until Receive Status = 0
Release Receive Buffer = 1; // empty RXFIFO
ResetRequest = 1 // enter reset mode

---->       /* ------------------------------ */
            // Error Passive Interrupt or
            /* ------------------------------ */
            AbortTransmission = 1; // abort a running transmission
do {
        Release Receive Buffer = 1; // empty RXFIFO
        } while (RBS == 1);
do {} while (TS); // wait until Transmit Status = 0
do {} while (RS); // wait until Receive Status = 0
Release Receive Buffer = 1; // empty RXFIFO
Reset Request = 1; // enter CAN reset mode
RXERR = 0; // Receive Error Counter = 0
TXERR = 0; // Transmit Error Counter = 0
Reset Request = 0; // back into operating mode
/* ------------------------------ */
// continue normal operation
```

偏差 2：空闲模式的定时器 T2 和 PWM

在该器件的空闲模式下发现一个异常情况。该异常情况下，空闲模式中定时器 T2 和两个 PWM 通道的功能被禁止。

所有其它 PP8xC591VFx/00 的节电模式都和数据手册所定义的一样。

描述

在空闲模式中，定时器 T2 计数器/定时器和它的预分频器以及 PWM1/PWM2 都没有复位或停止。这样就使得空闲模式所降低的功耗小于规定的值。

解决办法

要在空闲模式中降低功耗，定时器 T2 必须停止并复位，PWM0 和 PWM1 必须复位。

偏差 3：ESD 敏感度

按照人体模型，VDD 脚可耐受高达 1.5kV 的 ESD，而其它管脚可耐受超过 2kV 的 ESD。根据 maschine 模型，所有管脚都通过 250VESD 电压。

描述

VDD 脚的 ESD 耐受度最大为 1.5kV（根据人体模型）